Varieties of user-centredness: an analysis of four systems development methods

Juhani Iivari* & Netta Iivari†

*Department of Information Processing Science, P.O. Box 3000, 90014 University of Oulu, Oulu, Finland, email: juhani.iivari@oulu.fi, and †Department of Information Processing Science, P.O. Box 3000, 90014 University of Oulu, Oulu, Finland, email: netta.iivari@oulu.fi

Abstract. Based on an extensive literature analysis, this paper examines user-centredness in the context of systems development as a multidimensional concept composed of four aspects: user focus, work-centredness, user involvement and system personalization. Each dimension loads user-centredness with different meanings. The four dimensions can be used for evaluating systems development methods and approaches as to what extent and in what sense they adhere to the ideals of user-centredness. To illustrate this, the dimensions are applied to the analysis of four allegedly user-centred systems development methods: Goal Directed Interaction Design, Contextual Design, Scenario-Based Design and Human-Centred Systems Development Life Cycle. The analysis shows considerable variation in how these methods address the four dimensions of user-centredness.

Keywords: user-centred design, human-centred design, work-centred design, user involvement, user participation, personalization

1. INTRODUCTION

Since the early publications on user-centred design (UCD) (Knittle et al., 1986; Norman, 1986) user-centredness has aroused increasing attention in the context of information technology (IT) artefacts, leading, for example, to the ISO 13407 (1999) standard on the human-centred design of interactive systems. In this paper, we will use ‘user-centred’ and ‘human-centred’ as synonyms even though one can make a distinction between the first two (Gasson, 2003). Interestingly, the current focus on user-centredness has arisen at a time when UCD conditions have become challenging because of the market commodification of many IT systems and services and the increased use of application packages in organizational information systems (IS) development.

Despite all this interest, the actual concept of user-centredness has remained unclear (Carroll, 1996; Karat, 1997; Gulliksen et al., 2003; Kujala, 2003), partly because of the evolu-
tion of the idea of UCD, which has incorporated notions from different sources (Hellman, 1989), such as prototyping (Keen & Scott Morton, 1978; Floyd, 1984; Bødker & Grønbæk, 1991), evolutionary delivery (Keen & Scott Morton, 1978), socio-technical design (e.g. Bostrom & Heinen, 1977; Mumford, 1983; Pava, 1983; Eason, 1988), user participation (Mumford, 1983), participatory design (PD) (Greenbaum & Kyng, 1991; Schuler & Namioka, 1993) and usability engineering (UE) (Nielsen, 1993; Karat, 1997; Mayhew, 1999). More recently, the aforementioned challenges of UCD have stimulated researchers to propose new ideas such as representing users in terms of fictive users called ‘personas’ (Cooper, 1999) and introduction of various surrogates to represent users in the development process (Cooper & Bowers, 1995). These ideas differ so radically from some traditional ideas of UCD that it is appropriate to question in what sense they represent UCD at all. Altogether, UCD has acquired a variety of meanings from these various intellectual sources, which have led to alternative principles and techniques for supporting user-centredness.

The purpose of this paper is to analyse the variety of meanings of user-centredness in the existing literature. There are only a few published reviews of the history of user-centredness (Karat & Karat, 2003), and with the exception of Isomäki & Pekkola (2005), they all originated from the Human Computer Interaction (HCI) community and do not fully take into account IS research as a source of some of the ideas of user-centredness. It also seems that the IS research community has largely overlooked this research theme. In that sense, the present paper will fill a void in the current IS literature.

We propose here that user-centredness should be regarded as a multidimensional concept and will discuss how its dimensions have been approached in the UCD literature. These dimensions can be used for evaluating systems development methods and approaches as to what extent and in what sense they adhere to the ideals of user-centredness. To illustrate this, they are applied to the analysis of four allegedly user-centred systems development methods: the Goal-Directed Interaction Design (GDID) method (Cooper, 1999; Cooper & Reimann, 2003), the Contextual Design (CD) method (Beyer & Holtzblatt, 1998), the Scenario-Based Design (SBD) method (Rosson & Carroll, 2002) and the Human-Centred Systems Development Life Cycle (HCSDLC) model (Zhang et al., 2005; Te’eni et al., 2007).

2. AMBIGUITY OF USER-CENTREDNESS

The ISO 13407 (1999) standard defines human-centred design in terms of four principles: the active involvement of users and a clear understanding of user and task requirements; an appropriate allocation of function between user and system; iteration of design solutions; and multidisciplinary design. It is obvious that these principles are not very clearly formulated and do not define UCD unambiguously. One can argue, for example, that ‘a clear understanding of user and task requirements’ and ‘an appropriate allocation of function between user and system’ are more goals than principles to be followed. The UCD process may lead to these goals, but one often knows this only in hindsight, after the system has been implemented and used for some time.
Recognizing that the concept of UCD is ambiguously defined, Gulliksen et al. (2003) proposed 12 principles for UCD: user focus; active user involvement or participation; evolutionary, i.e. iterative and incremental, system development; simple design representations, understandable to users and other stakeholders; early and continuous prototyping; evaluation of use in context; explicit and conscious user interaction/interface design activities; a professional attitude; usability; holistic design; process customization locally; and a user-centred attitude. These principles obviously define UCD more clearly even though it is questionable whether all 12 need to be included. Principle 9, for example, requires that ‘the usability designer must be given the authority to decide on matters affecting the usability of the system and the future use situation’ (p. 403). Since Gulliksen et al. (2003) interpreted usability as ‘the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction, in a specified context of use’ (ISO 9241-11, 1998), which also covers usefulness or utility (pp. 401, 407), it is not realistic to expect that any usability designer or usability group could have such a broad authority as it would mean that all stakeholders related to the system would delegate their decision-making power to the usability designer or usability group.

Furthermore, there seems to be ambiguity regarding the overall goal of UCD. The ISO standard 13407 defines UCD as ‘an approach to interactive systems development focusing specifically on making systems usable’. Unfortunately, usability is a quite ambiguous concept as Seffah and Metzker (2004) noted. It can be interpreted as embracing usefulness or utility as well (e.g. in Rosson & Carroll, 2002; Gulliksen et al., 2003), while some allegedly ‘user-centred’ methods explicitly define their goals to include usefulness in addition to usability (e.g. Beyer & Holtzblatt, 1998; Cooper, 1999). The ISO 13407 standard for UCD has also been criticized for providing very ambiguous guidance on designing ‘usability’ so that refined guidelines concentrating specifically on usability have had to be developed (e.g. Jokela, 2002; Jokela et al., 2003). This jungle of ‘user-centred design’ will be analysed in more detail later.

3. ANALYSIS OF THE VARIETIES OF USER-CENTREDNESS

A literature search using the key words ‘user-centered design’, ‘human-centered design’, ‘customer-centered design’, ‘user-centeredness’, ‘human-centeredness’ and ‘customer-centeredness’ was carried out using a number of bibliographic databases [ABI/INFORM (ProQuest), Academic Search Premier (EBSCO Host), ACM Digital Library, Elsevier Science Direct, IEL Online]. To keep the number of hits within a reasonable limit, the keywords were applied to abstracts rather than the full texts of scholarly journals. After deleting false positive hits, 347 relevant papers were identified, the yearly distribution of which, starting from 1988, is shown in Table 1. There was just one reference before 1988 that had one of the aforementioned keywords in its abstract. Table 1 shows that interest in human-centred design/UCD started to grow in the mid-1990s, achieving momentum in the 2000s. One should note, however, that the topic has been of particular interest in practitioner-oriented magazines such as Interactions. This latter journal accounted for 42 hits, i.e. 12% of the total of 347, a
proportion that grew to 23% in 2004–2007. A search covering the years 2008 and 2009 showed that the number of papers on user or human-centredness has remained high, with dozens of papers published in journals each year. A quick scan of these papers suggested, however, that they do not add anything essentially new from the viewpoint of the present discussion, and therefore, they were not included in the more systematic review. Inspired by Beyer & Holtzblatt (1998) and Liang and Tanniru (2006–2007), we also experimented with the keyword ‘customer-centric design’, but it did not lead to any hits.

The 347 papers form the major body of the literature that was used here to inductively identify the dimensions of user-centredness. The purpose of the present paper is not to provide a systematic literature review in the sense of classifying each of the 347 papers. Many of them do not attempt to conceptualize ‘user-centredness’ but assume it to be a self-explanatory concept, or else they refer to authorities such as Norman (1986; 1988), Gould & Lewis (1985) or the ISO 13407 standard, or simply use user-centredness as a contrast to designer/system/technology-centredness. A number of the papers are case studies characterizing the way in which a system was built in order to be human/user-centred.

The particular focus in the present paper will be on those papers that conceptualize, define, exemplify and/or problematize the concepts and dimensions of ‘human-centred design’, ‘user-centred design’, ‘customer-centred design’, ‘human-centredness’, ‘user-centredness’ and ‘customer-centredness’. The analysis enabled four dimensions of user-centredness to be identified:

<table>
<thead>
<tr>
<th>Year</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1988</td>
<td>1</td>
</tr>
<tr>
<td>1989</td>
<td>3</td>
</tr>
<tr>
<td>1990</td>
<td>5</td>
</tr>
<tr>
<td>1991</td>
<td>6</td>
</tr>
<tr>
<td>1992</td>
<td>7</td>
</tr>
<tr>
<td>1993</td>
<td>7</td>
</tr>
<tr>
<td>1994</td>
<td>8</td>
</tr>
<tr>
<td>1995</td>
<td>15</td>
</tr>
<tr>
<td>1996</td>
<td>12</td>
</tr>
<tr>
<td>1997</td>
<td>11</td>
</tr>
<tr>
<td>1998</td>
<td>13</td>
</tr>
<tr>
<td>1999</td>
<td>27</td>
</tr>
<tr>
<td>2000</td>
<td>17</td>
</tr>
<tr>
<td>2001</td>
<td>36</td>
</tr>
<tr>
<td>2002</td>
<td>26</td>
</tr>
<tr>
<td>2003</td>
<td>30</td>
</tr>
<tr>
<td>2004</td>
<td>39</td>
</tr>
<tr>
<td>2005</td>
<td>52</td>
</tr>
<tr>
<td>2006</td>
<td>38</td>
</tr>
<tr>
<td>2007</td>
<td>42</td>
</tr>
</tbody>
</table>
1 user-centredness as user focus;
2 user-centredness as work-centredness
3 user-centredness as user involvement or participation; and
4 user-centredness as system personalization.

With the possible exception of the fourth point, there is nothing surprising in these dimensions. Hall (2001), for example, characterized a user-centred approach as one that ‘involves knowing who the users will be, their capabilities, needs and expectations, their goals and the tasks required to achieve those goals, and the physical and social environments in which users have to achieve those goals. It involves processes of PD, user testing and iterative design’ (p. 486). The value of the following analyses lies in making explicit a variety of options and views regarding user-centredness in each of the four dimensions. The relevance of system personalization as a dimension of user-centredness will be explained later.

At the same time, Hall’s (2001) characterization pointed out that the four dimensions do not form an absolute ‘truth’ regarding user-centredness. For example, his characterization, similar to almost all definitions of user-centred design, includes iterative design (e.g. ISO 13407 standard). We have excluded it since we do not see it as inherent to user-centredness in the sense that iterative design alone can make systems design user-centred. Iterative design, such as prototyping, only provides an effective means of involving users in the process.

The following analysis will proceed on the aforementioned four dimensions. One should note, however, that the detailed analysis and discussion of the dimensions is not confined to the 347 papers identified but may refer to the literature outside these in order to expose some of the intellectual roots of user-centredness that may have remained unrecognized in previous accounts of UCD.

3.1. User-centredness as user focus

User-centredness as user focus represents the original emphasis of UCD. Norman (1986; 1988), for example, who coined the term ‘user-centred design’, put forward seven principles: use both knowledge in the world and knowledge in the user’s head; simplify the structure of tasks so that user’s short-term and long-term memories are not overloaded; make things visible to bridge the gulf of execution and evaluation; get the mapping right; exploit the power of constraints, both natural and artificial, in order to give the user the feel that there is one thing to do; design for error to allow the user the option of recovery from any possible error made; and when all else fails, standardize. User-centredness as user focus has been visible more recently in characterizations such as ‘User-Centered Design (UCD) is a process that places the person at the centre, focusing on cognitite factors (such as perception, memory, learning, problem-solving, etc.) as they come into play during people’s interactions with the interface’ (Shahar et al., 2006, p. 119).

More generally, one may claim that the ideal of UCD is to take every individual user’s capabilities into consideration and fully satisfy his or her needs related to the system to be developed. These needs may be determined by the activity or work to be supported by the
User-centredness as user focus leads to the question ‘how can we identify and represent users?’ When the system has one or only a few users, it is fairly easy to identify them and to match the system to their needs. Decision support systems (DSS) (Keen & Scott Morton, 1978) illustrate the case. A DSS is typically assumed to have one or only a few users, and Keen & Scott Morton (1978) emphasized in several contexts that a DSS must be tailored to match the decision-maker (user) (pp. 6, 11, 50, 58). This matching was supported by a highly iterative development process, including both prototyping and evolutionary development (Keen & Scott Morton, 1978), which were also used to characterize UCD (e.g. Gulliksen et al., 2003). By a prototype, we refer to an experimental system that attempts to illustrate certain aspects of a ‘final’ system (e.g. the user interface) and is intended to be used only experimentally, whereas the system versions in evolutionary development are fully operational and intended for real use (Iivari, 1982). The need for evolutionary development in the DSS context arose from the fact that a DSS tended to change the decision-maker’s (user’s) decision-making process so that it was necessary to adapt the system accordingly to fit the individual user.

The aforementioned ideal of satisfying each user’s individual needs exactly is difficult to achieve for two major practical reasons. First, many systems are used for communication, coordination, cooperation and control of distributed activities carried out by a number of users. In this case, individual users cannot expect their needs and preferences to be fully satisfied but must be prepared to make compromises in the interests of the uniformity and compatibility of the system. This is especially true when developing systems of an infrastructure type. Second, a system may have a huge number of globally distributed users, many of whom cannot be accessed when designing the system. Accordingly, the system may be designed for an average or fictive user.

When it is impossible to analyse the needs of all users individually, one strategy is to identify human factors that allow general principles or guidelines for design to be derived. The early publications on ‘user-centredness’ based on psychology and ergonomics (Knittle et al., 1986; Norman, 1986) reflect this strategy. Dillon & Watson (1996) recommended that users should be analysed in terms of individual differences identified in psychology over the last 100 years in order to characterize users in UCD, and Allen (2000) and Sun & Ousmanou (2006) are more recent examples of this aspect of user-centredness.

Recognizing that users are heterogeneous (e.g. Knittle et al., 1986), one strategy is to segment the actual or potential user population so that the users within each segment can be expected to be relatively homogeneous and to design a different version of the system for each segment. This segmentation allows more and more detailed user profiles to be derived empirically, but however detailed it is, this strategy will always lead to a focus on the average or typical user within each segment.

More recently, Cooper (1999) put forward a provocative suggestion that systems design should be based on ‘personas’ rather than real users. A persona is a fictive user – a hypothetical archetype for actual users – and can be given a name and a face and be carefully
described in terms of needs, goals and tasks. Personas will be discussed in more detail in section 4, where GDID (Cooper, 1999; Cooper & Reimann, 2003) is evaluated in more detail.

### 3.2. User-centredness as work-centredness

Bannon (1991) criticized the human factors strategy as follows: ‘Although psychology, particularly as represented by the field of human factors (HF), or ergonomics, has had a long tradition of contributing to computer systems design and implementation, it has often neglected vitally important issues such as the underlying values of the people involved and their motivation in the work setting’ (p. 26). He proposed an alternative perspective of understanding the relationship between people, technology, work requirements and organizational constraints in work settings, where people are actors in situations, with a set of skills and shared practices based on their experience of working with others. This has led to a significant turn in HCI research. Understanding users’ work activities and the context of use has become a central focus of HCI research, and also of UCD as represented in the scientific literature (e.g. Klein et al., 1997; Gulliksen et al., 2003; Scott et al., 2005; Thursky & Mahemoff, 2007) and in the more practitioner-oriented literature (e.g. Beyer & Holtzblatt, 1998; ISO 13407, 1999; Mayhew, 1999; Rosson & Carroll, 2002).

The focus on the users’ work domain raises two new questions: ‘how can one conceptualize and represent the work domain?’ and ‘what are the drivers of changes in the work domain?’ Related to the former question, one can identify the problem of representing local work practices vs. holistic modelling of the work domain. In the case of drivers of change, one can distinguish technology-driven, work process-driven, interactive or emergent changes.

Historically, the question of how to conceptualize the work domain related to a computer system has been under focus in the context of office automation, computer-supported cooperative work (CSCW) and IS research more generally. Socio-technical thinking has been especially influential in IS research (e.g. Bostrom & Heinen, 1977; Mumford, 1983; Pava, 1983; Eason, 1988). Research into CSCW has focused especially on the detailed work practices of users in cooperative settings (Schmidt & Bannon, 1992). In addition to socio-technical design theory, structuration theory (e.g. Jones & Karsten, 2008), actor network theory (e.g. Hanseth et al., 2004) and activity theory (e.g. Redmiles, 2002) have been proposed more recently in order to provide a theoretical understanding of the relationship between work and IT.

The work domain of many complex systems such as enterprise resource planning (ERP)-based IS covers several organizational functions. Note, however, that ERP packages are not IS as such but only software that can be used to implement IS. When considering the development of such a complex system, holistic design (Gulliksen et al., 2003) implies that the information system must be developed in parallel with the whole work domain. Suchman (1995) noted that one of the benefits of job specialization is that we can ‘black box’ the work of others so that we do not need to know how they get their work accomplished, although we are at the same time dependent on their products and services. This means that perhaps no single individual will have an understanding of the whole work domain affected by the system.
In order to be able to discuss the work domain in holistic terms, it should be made visible in some way using appropriate representations.

Representations always include abstraction, decisions as to what to include and what to leave out. Gulliksen et al. (2003) claimed that design representations in UCD should be simple and easily understandable for users and other stakeholders. Bearing in mind the variety of potential users, it is not always so obvious what sorts of representations are understandable to users. A representational formalism that is meaningful to one group may be totally incomprehensible to another (e.g. professional electrical engineers may differ from supermarket cashiers in this respect). CD (Beyer & Holtzblatt, 1998), with its five work models – flow model, sequence model, artefact model, cultural model and physical model – is an excellent example of work-centredness in the UCD literature. This will be analysed in more detail in section 4.

Regarding the drivers of change and the consequences of IT for the work domain, the UCD literature does not discuss in detail assumptions about how IT allows new forms of working. Similarly, to the four perspectives suggested by Gallivan & Srite (2005) to describe the relationship between IT and culture (technological determinism, the organizational imperative, the interactionist view and the emergent view), we also distinguish four views to describe the relationship between technology and the work domain. The first standpoint is to regard technology as the primary motor of change (the technology-driven view). It can then be assumed that the introduction of technology will change the work domain. Norman (2005), for example, advocated activity-centred design rather than UCD, pointing out that for many activities, the tools define the activity. The opposite standpoint is to take changes in the work domain as primary (the work process-driven view) and to consider first how to change business processes [e.g. change analysis in Lundeberg et al. (1981) and soft systems methodology, Checkland & Scholes (1990)]. Socio-technical design represents an intermediate position (the interactive view), suggesting that the social subsystem and the technical subsystem should be designed in parallel in order to achieve joint optimization. The opinion that the adaptation of technology and work is more of an emergent process (the emergent view) has gained acceptance more recently (Markus & Robey, 1988; Markus et al., 2002), and the interpretive flexibility of IT artefacts has been recognized rather than technological determinism (e.g. Orlikowski & Gash, 1994; Orlikowski, 1996; Grint & Woolgar, 1997; Robey & Boudreau, 1999). It has also been acknowledged that no matter how well-designed and implemented it may be, IT may produce unexpected, paradoxical or ironic consequences in organizations (Robey & Azevedo, 1994; Robey & Boudreau, 1999).

3.3. User-centredness as user involvement

Most views of UCD consider active user involvement to be an integral part of it (ISO 13407, 1999; Gulliksen et al., 2003), although Carroll (1996) noted that UCD can also be non-participatory. There is a long tradition of research into user participation and involvement, especially in IS research (see Kujala, 2003; Markus & Mao, 2004 for fairly recent reviews). Barki & Hartwick (1989) suggested a conceptual distinction between ‘user participation’ and ‘user involvement’ where participation refers to assignments, activities and behaviours that users or their repre-
sentative perform during the systems development process, and involvement to a psychological state reflecting the importance and relevance a user attaches to a given system. Although this is a clear and justified distinction, it may be difficult to interpret user participation to cover weak forms of ‘participation’ where users may be just passive objects of observation.

Several authors (e.g. Grudin, 1991; Kyng, 1994; Kujala, 2003; Iivari, 2006a) have pointed out that user participation is challenging especially in a product development context, where it is often difficult to identify and make contact with prospective users. One is often obliged to be satisfied with weaker forms of user ‘participation’ where some of the users are just observed, interviewed, represented or merely surrogated. In this situation, we prefer to speak about ‘user involvement’ as a broader concept in which users are somehow involved in the systems development process. ‘User participation’ is a special case of user involvement in which users (or their representatives) actively participate in the process. This terminology, although contradictory with Barki & Hartwick (1989), seems to be in line with the literature on user involvement in product and service development contexts (e.g. Alam, 2002; Kujala, 2003).

User-centredness as user involvement leads to the question of ‘why should users be involved?’ Much of the literature on user involvement has focused on users’ involvement in the development of workplace systems. This involvement has typically been hierarchically governed, whereas user involvement in product development is more of a market-driven issue, dependent on the degree to which companies that produce IT artefacts or services see user involvement in order to understand users’ needs as vital for their business success. Generally speaking, user involvement in systems development, especially in the workplace context, can be justified in terms of democratic empowerment, which maintains that users should be empowered to participate in decision-making in their workplace, or functional empowerment, which maintains that the users should be empowered to do their job effectively and efficiently (Clement, 1994). At the same time, we note that ‘user empowerment’ is a complex issue. Its critics maintain that user empowerment is often more management rhetoric than serious redistribution of power in organizations, claiming that management may use it as a means for motivating the employees to strive for management goals by giving them some power (O’Connor, 1995; Hardy & Leiba-O’Sullivan, 1998; Howcroft & Wilson, 2003). These critics contend that empowerment cannot occur through those having power giving some to others but that it should be seen as a means to fight against different forms of social domination, i.e. the oppressed combating the oppressors.

The UCD literature (e.g. Beyer & Holtzblatt, 1998; Mayhew, 1999; Rosson & Carroll, 2002) often refers to the Scandinavian trade unionist approach when speaking about user involvement. Democratic empowerment of users was clearly the original goal of the Scandinavian trade unionist approach (e.g. Bjerknes et al., 1987; Greenbaum & Kyng, 1991), although this emphasis decreased in its later adaptations to the PD stream of research (Schuler & Namioka, 1993). The functional empowerment of users, on the other hand, i.e. their empowerment through providing them with useful and usable solutions, is evident in the UCD literature covering both workplace and product development contexts. The emphasis may be solely on achieving management goals, however (especially Beyer & Holtzblatt, 1998, according to Spinuzzi, 2002), which is in stark contrast with the original aims of the Scandinavian approach.
A second question is ‘how should users be involved?’ Damodaran (1996) classified users’ roles in the design process as informative, consultative or participative. In the informative role, users act as providers of information and as objects of observation, in the consultative role, they are allowed to comment on predefined design solutions, while in the participative role, they actively take part in the design process and have decision-making power regarding the design solution. Here, again the UCD literature differs considerably from its origin, the Scandinavian trade unionist approach. Whereas the latter has advocated active user participation and cooperative design in which people with different competences appreciate each other and jointly create new work practices and technologies (Greenbaum & Kyng, 1991; Schuler & Namioka, 1993; Kensing & Blomberg, 1998; Asaro, 2000; Beck, 2002; Spinuzzi, 2002), the HCI literature is often satisfied with informative and consultative user involvement (Iivari, 2006b).

User involvement can also be divided into direct and indirect involvement. Users may be directly involved in an informative, consultative or participative role. If an information or software system has a large number of prospective users, it is practically impossible to rely on direct involvement, especially when the users are not known in advance (as is often the case in an IT product development context). In this situation, recourse may be to have indirect user involvement through a representative or surrogate.

In representative user involvement, a small set of users may be selected or elected to represent the whole user population (Mumford, 1983). The representation may be ‘political’ in the sense that the representatives are elected in some way by the users, or ‘statistical’ in the sense that they are typical users, possibly from different user categories. In both cases, even though these user representatives may really represent the whole user population at the beginning, there is also a risk, especially if their participation is active and long-lasting, covering the whole development project that after some time, they will no longer be truly representative of the users (Hedberg, 1975).

In surrogate user involvement, intermediaries are chosen to stand for the real users (cf. Cooper & Bowers, 1995). These may be UCD/UE/usability specialists, for example. They do not ‘represent’ the users in the same sense as was implied earlier, i.e. the users have not elected them as representatives nor are they typical users. They are just assumed to have, or to be able to acquire, an understanding of the users and their work practices and to contribute this understanding to the design (Iivari, 2006a). In this case, user involvement can be informative or consultative at the most (Damodaran, 1996; Iivari, 2006b).

It has been claimed that the ultimate form of user involvement is a situation where the users themselves design and implement their systems (e.g. Järvinen, 1982). There exists research in the area of End User Computing (EUC) (Brancheau & Brown, 1993; Downey, 2004), and more recently, in End User Development (EUD) (e.g. Sutcliffe & Mehandjiev, 2004). Although there is no agreement as to what EUC actually is (Downey, 2004), in our interpretation, it is particularly focused on the development of systems for personal use. The more recent term ‘End User Development’ refers very widely to personalizing and customizing applications, programming them and designing them without seeing the underlying programming code (Fischer et al., 2004; Sutcliffe & Mehandjiev, 2004). Although EUC/EUD can be regarded as one ultimate form of user-centredness, it has not been discussed much in the UCD literature,
where a certain division of labour between developers and users is normally assumed except for the issue of personalization and customization, which will be discussed in section 3.4. Very recently, Keinonen (2009) has categorized UCD approaches on the basis of how much users and designers contribute to the design, identifying, among other things, a ‘do-it-yourself design’ approach in which the users are proactive while the designers remain inactive. He mentions open source software (OSS) development (e.g. Franke & von Hippel, 2003) and Fischer’s meta-design concept (see Fischer et al., 2004), which is connected with the EUD tradition, as concrete examples of this kind of UCD. Differing from Keinonen (2009), we are more wary of considering the OSS development a form of UCD since the former does not need to be associated with personal use of the system, and the very distinction been ‘designers’ and ‘users’ may be very difficult to draw in the OSS development, the key developers usually being qualified designers or programmers.

User involvement also has a power dimension: ‘who has the power and right to decide about changes in the system and in the work domain?’ In view of the influence of the Scandinavian trade unionist approach on UCD, it is amazing that power and politics has been almost forgotten in the UCD literature. Focusing mainly on systems that are developed for work contexts, prior research has argued that there is a potential for conflicts between management and workers (Bjerknes et al., 1987; Gärtner & Wagner, 1996; Nielsen, 1999; Howcroft & Wilson, 2003), developers and users (Kirsch & Beath, 1996; Nandhakumar & Jones, 1997; Symon, 1998), and different organizational units and occupational groups (Markus, 1983; Ehn, 1988; Butler & Fitzgerald, 1997; Symon, 1998). Although the UCD literature seems to assume that the users should have the power of decision regarding the solutions adopted, we cannot forget the political milieu in which the systems development takes place (Garretty & Badham, 2004). Taking into consideration the power relations prevailing in organizations, it is not realistic to assume that users alone should have the right to decide about changes in the work domain, including the IT shaping it. One can assume that at least management should have its say, too. Users form only one stakeholder group in systems development. Finally, some researchers assign a considerable power of decision in matters related to users’ work practice design to the UCD/UE/usability specialists who are assumed to take part in the systems development (Iivari, 2006a).

3.4. User-centredness as personalization

The diversity of users, the difficulty of involving them in the design process, especially in a product development context, and users’ learning processes as they gain experience in using the system all make it difficult to design systems to fit every user. Franke & von Hippel (2003), for example, noted that customers (and users) of a given type of product or service can have quite different needs. Even though the market may be segmented, the product targeted to each segment addresses only the average or typical customers’ needs in that segment. They also point out that recent technological advances have reduced the cost of designing and implementing products for ‘markets of one’ and propose ‘innovation toolkits’ to enable users to modify the systems on their own.
Users have also been found to have a tendency to appropriate and adapt the solutions developed for them to fit their own work practices, which is typically a continuous, evolving process (e.g., Keen & Scott Morton, 1978; Dourish, 2003; Bansler & Havn, 2004; Pipek, 2005). This has led a number of researchers to suggest adaptable, user-customizable (Dourish, 2003) or user-tailorable (MacLean et al., 1990; Germonprez et al., 2007) technologies so that they can easily be adapted to a user’s particular needs, activities or setting. The EUD literature has emphasized that EUD can be enabled by making software smarter and by allowing easier configuration and customization by the users (Sutcliffe & Mehandjiev, 2004). It is also suggested that systems should be ‘underdesigned’, meaning that complete solutions should not be aimed at during the development, but instead, instruments should be provided for users to modify, customize and extend the solutions to fit their own needs (Fischer et al., 2004). This comes close to the idea of personalization and personalizable systems that have aroused some interest in the context of UCD (Kramer et al., 2000; Albert et al., 2003; Karat et al., 2003; Petrie et al., 2005; Streefkerk et al., 2006; Liang & Tanniru, 2006–2007).

A personalized system adapts or allows to adapt the system’s content structure, presentation form and functionality to each user’s characteristics, use behaviour and usage environment (Kobsa et al., 2001). Personalization may be achieved by designing the system to be adaptable or adaptive (Brusilovsky, 1996). Adaptability allows the user to change the system according to his or her preferences, and adaptivity means that the system automatically adapts to the user (Brusilovsky, 1996; Frias-Martinez et al., 2005). Adaptivity means that the system builds a user model and automatically adapts to the accumulated user model (Brusilovsky, 1996; Frias-Martinez et al., 2005). User models can be created using either a user-guided approach or an automatic approach. In the former, the user explicitly provides the information. In the latter, the user model is created by the system by observing the user’s usage patterns.

3.5. Summary

Based on the analysis of the extant literature on UCD, we identified four dimensions in user-centredness: user-centredness as a focus on the user, user-centredness as work-centredness, user-centredness as user involvement and user-centredness as system personalization. We also found a number of options in the ways these are addressed in the UCD literature (Table 2).

Table 2 does not take into account the relationships between the dimensions of user-centredness. Norman (2005), for example, contrasted ‘human-centred design’ with ‘activity-centred design’, the latter representing an extension of work-centredness (see section 5). Although he does not see them as mutually exclusive, he pointed out that excessive focus on individual users may detract from support for the activities and lead to a lack of cohesion and added complexity in the design. This has obvious implications for user involvement. Norman (2005) explicitly noted that sometimes, the designer should have absolute authority to decide on the design. In this situation, potential user involvement is functional and is likely to be
informative or consultative at most but hardly participative. At a more detailed level, one can argue that if the user focus is limited to a fictive or average user, user involvement is necessarily restricted to being informative, in the sense that users act only as providers of information or objects of observation. ‘Personas’ and user profiles are based on data gathered concerning potential users.

4. ANALYSIS OF USER-CENTREDNESS IN FOUR SYSTEMS DEVELOPMENT METHODS

As discovered earlier, user-centredness is a rich, multidimensional concept. It is therefore interesting to try to understand to what extent and in what sense it is covered in systems development methods that claim or are claimed to be user-centred or human-centred. The purpose of this section is to analyse user-centredness in four fairly recent systems development methods:

1. GDID (Cooper, 1999; Cooper & Reimann, 2003);
2. CD (Beyer & Holtzblatt, 1998);
3. SBD (Rosson & Carroll, 2002); and
4. HCSDLC (Zhang et al., 2005; Te’eni et al., 2007).

The four methods were selected to represent the variety of backgrounds to be found in ‘user-centred’ methods. They all have strong roots in HCI, but the HCSDLC model (Zhang et al., 2005) in particular also has a notable IS origin. The purpose of the whole method is to integrate HCI concerns into the IS development process. Interestingly, CD (Beyer & Holtzblatt, 1998), which originates from the HCI community, is relatively silent about HCI design issues, its main emphasis being on the analysis and redesigning of the working context of the future system.

© 2010 Blackwell Publishing Ltd, Information Systems Journal 21, 125–153
4.1. Findings regarding the systems development methods

The GDID Method

The GDID method (Cooper, 1999; Cooper & Reimann, 2003) describes interaction design for producing power and pleasure for users. Interaction design includes definition of the behaviour, functions and information needed and their presentation to the users (Cooper, 1999). Cooper (1999) identified three phases in the method: conceptual design (what is valuable for users), behavioural design (how the system behaves) and interface design (what it looks like). Cooper & Reimann (2003) revised the method to include five phases: research, modelling, requirements definition, framework definition and refinement. The first three phases are included in the conceptual design phase in Cooper (1999), whereas the last two represent the last two phases.

GDID has a strong user focus but it is on fictive users (personas). One begins with an empirical investigation. Actual users are ‘a valuable resource’ as ‘raw data’, but they are considered ‘useless for the design process’ since they have ‘anomalies’ and ‘idiosyncrasies’, and ‘are not representative’ (Cooper, 1999, p. 123). Instead, one should construct ‘personas’ that are pretended users, ‘hypothetical archetypes of actual users’ defined with rigour and precision (Cooper, 1999, p. 124). Cooper & Reimann (2003) had developed their personas to resemble user profiles by placing more emphasis on the empirical inquiry part. The method is also oriented towards human factors (human constraints, limitations, needs, strengths and potentials). Cooper (1999) discussed extensively the characteristics of programmers and users that cause problems for interaction design, the general human (personal) and corporate goals that users need to be able to achieve, and several interaction design principles (e.g. software needs to be polite, it needs to be similar to a sensitive and caring friend). Cooper & Reimann (2003) argued that interaction design is about understanding users, cognitive principles and human motivation, and discussed in length interaction design principles and patterns, and the principles of visual interface and information design. Neither book, however, contains much discussion of the theories behind these issues.

At the same time, GDID has a noticeable work orientation. Cooper (1999) did not discuss users’ current practices, but it is assumed that they are analysed during the empirical investigation. The main effort is put on the redesign of users’ work practices. Cooper & Reimann (2003) devoted the research phase to the identification of usage patterns, which together with the associated goals, help to create the personas. Then, during the modelling phase, the usage and workflow patterns that have been identified are synthesized into domain and user models (pp. 16–17). Flow, sequence, artefact and physical models of the CD (Beyer & Holtzblatt, 1998) are also presented. Both GDID books also discuss work redesign in length. It is argued that interaction design needs to make sure that the users can achieve their goals. Users’ tasks are designed during the scenario construction phase, given that a scenario is ‘a concise description of a persona using a software product to achieve a goal’ (Cooper, 1999, p. 179). The main focus seems to be on the redesign of users’ local work practices (tasks), but Cooper & Reimann (2003) also acknowledged holistic work models.
The view of change in GDID is technology-driven or interactive. Engineering or marketing-driven development is criticized. Interaction designers need to have the authority to decide on what to include in the design – they should be allowed to be the ‘ultimate owners of the product quality’ (Cooper, 1999, p. 232). Engineering defines ‘what is capable’, business defines ‘what is viable’, but interaction design should define ‘what is desirable’, which needs to be acknowledged in order to succeed (Cooper, 1999, p. 73).

GDID advocates user involvement in a very limited sense. Users are to be interviewed or observed, the fictive personas are assumed to be present in the design meetings and discussions, but users should not be allowed to affect the solutions. As a consequence, the method favours surrogate user involvement: interaction designers are positioned as user advocates responsible for designing for power and pleasure for users.

Although GDIS recognizes power issues only to a weak extent and users are given no power, both Cooper (1999) and Cooper & Reimann (2003) demanded more power for interaction designers in relation to engineers/managers. Actual users are a ‘valuable resource’, but one should never let them ‘directly affect the solution’ (Cooper, 1999, p. 123). Cooper & Reimann (2003) also acknowledged that interaction designers should take the organizational culture and the level of security, etc., in the user organization into account and bear all relevant social and political issues and ethical questions in mind.

GDID recognizes personalization as an issue and discusses it in some length from the viewpoint of both adaptivity and adaptability. It is mentioned that software can be configured for different users, and each primary persona should have a unique interface. Inflecting the interface is also recommended: only frequently used functions should be offered as defaults. Cooper (1999) presented principles that argue for making software more adaptive and polite (‘considerate’ in Cooper & Reimann, 2003). Cooper & Reimann (2003) devoted a whole chapter on how to make software smart. In addition, they discuss in another chapter the question of allowing personalization and configuration in order to design for different needs (Cooper & Reimann, 2003).

The CD Method

CD (Beyer & Holtzblatt, 1998) is a systems development method that puts the understanding and design of customers’ work practices in the forefront. The method maintains that the goal is not primarily to design technology but to design a new way of working that is supported by technology. The method is composed of six phases: contextual inquiry, work modelling, consolidation, work redesign, system design and prototyping. Customer data gathered during the contextual inquiry forms ‘the base criteria for deciding what the system should do and how it should be structured’ (p. 3). The method covers the defining of both functionality and human–computer interaction, but user interface design is addressed rather weakly in this method, and it is emphasized that other methods cover it more extensively.

CD argues that one must first ‘understand the customers: their needs, their desires, their approach to the work’ (p. 22). During the contextual inquiry, individual users are ‘interviewed’, and individual work models are produced on the basis of the findings. The word ‘interview’ is
a misnomer in this context since contextual inquiry relies on observing users as they do their work and talk about it while doing it rather than interviewing, a process that is controlled by the interviewer. As a consequence, a user is, in the ideal case, an anonymous performer of the work observed by the ‘interviewer’. The method does not directly pay attention to the individual user, to his or her characteristics and preferences, but rather, individual users are taken into account only indirectly. Furthermore, it is argued that one needs during the consolidation phase to concentrate on the common structure and pattern inherent in the work accomplished by different people (p. 23).

The method has a strong work orientation. A detailed understanding of customers’ current work practices needs to be gained, and the individual work models include a flow model outlining communication and coordination, sequence models for each task, artefact models for the relevant artefacts, and cultural and physical models describing the influences and constraints imposed by the cultural and physical environments. Consolidated work models are then produced, revealing the common patterns and structure. Later, the method introduces work redesign as a distinct step in the development process. Visions are produced to describe the redesigned work practice as a coherent whole, and storyboards to describe instances of doing the redesigned work. In all, one can conclude that the method emphasizes both local work practices and holistic work models.

Change seems to be primarily work practice-driven; the design team should have some knowledge of what can be enabled by technology, but the real invention concerns the customer’s new work practice. Interactive and emergent views are also acknowledged. Technologies transform work tasks over time, and this process is unpredictable. People may invent new ways of using the system.

CD advocates some user involvement, depending on the nature of the system and the phase of system design. In the case of internal systems development, customer representatives may be in the design team (p. 248), but when the method is used to develop systems for external use, the customers is not assumed to participate in the design team. Customers (users) are observed and interviewed during the contextual inquiry, but they are involved mainly as objects of observation and providers of information, although they may also serve as co-interpreters of the work models. After the contextual inquiry, the design team makes all the design decisions without any active user participation. Users are invited to participate again in the prototyping phase when they are allowed to comment on predefined design solutions in recognition of the fact that ‘the customers are the final arbiters of what works and what doesn’t’ (p. 368). When developing an internal system, the CD team can also present the work models to the customers and redefine responsibilities, roles and procedures together with the internal customers. PD may be used in this phase (p. 428).

Questions of power are addressed to some extent in the method. During the contextual inquiry, one is supposed to gather data about cultural and political influences and constraints for the cultural model, and the consolidated cultural model is used as a reminder of these issues during the work redesigning phase. The design team can consider the various design options from the viewpoint of power relationships and pervasive values. More power is demanded for CD teams, which are expected to be multidisciplinary teams that may also
include customers if the system is being developed for internal use. People from different departments can represent their interests in the design team, but the users are given little power of decision in the method as the CD team makes the design decisions based on customer data and is responsible for innovations and for the resulting corporate response. ‘Customer data never dictates exactly what to design’ (p. 273).

CD recognizes personalization as an issue but does not really address it. It is argued that a system needs to address a customer population but must also allow for individual variation and should not sacrifice flexibility. The system may need to recognize and support different strategies for approaching a task (p. 261). The style of the user interface should be tailored to fit the user, and the data should be tailored to fit the needs of the user. The method does not, however, discuss in any greater depth how to make the system more adaptive or adaptable.

The SBD Method

SBD (Rosson & Carroll, 2002) is a UE method for developing human–computer interaction with the help of scenarios. SBD is composed of six phases: requirements analysis, activity design, information design, interaction design, prototyping, usability evaluation and documentation design. The method covers the defining of functionality as well as information and human–computer interaction design.

SBD has a strong user focus, specifically on empirical user profiles, fictive users and human factors. The method begins with empirical investigation. Requirements analysis considers multiple stakeholders, who are to be observed and interviewed. The resulting summaries take the form of stakeholder profiles and stakeholder diagrams showing the relations between the stakeholders. Also, summaries of artefact and task analyses are produced. Scenarios are then constructed, which include actors, i.e. hypothetical stakeholders that are imaginary but typical (in the sense of Cooper, 1999). The method also has a clear human factor orientation. HCI knowledge related to human perception and interpretation, gestalt principles and making sense of the information, for instance, is expected to be used in connection with activity, information and interaction design.

At the same time, SBD has a strong work orientation. After the empirical analysis, scenarios are constructed. A scenario, ‘a story about people carrying out an activity’ (p. 2), may include a setting, actors, task goals, plans, evaluation, actions and events (p. 18). Problem scenarios are situated in the problem domain, whereas design scenarios describe new visions (p. 2). The problem scenarios describe important characteristics of the users, their typical and critical tasks, the tools they use and their organizational context in the problem domain, whereas the activity design scenarios are narratives of ‘typical or critical services that people will seek from the system’ (p. 26), focusing on functionality. Information design scenarios offer details about the information provided, and interaction design scenarios offer details of human–computer interaction and feedback mechanisms. These information and interaction design scenarios can be refined with storyboards and sketches. This means that both current and future work practices are emphasized. In addition, the method acknowledges both local work practices and holistic work models, although the emphasis seems to be on the former, whereas the latter is
mainly mentioned. It is mentioned that collaborative goals and needs have to be acknowledged in the scenarios and that stakeholder diagrams are to be produced.

The view of change in the SBD method seems to be interactive. The aim is to design effective, comprehensible and satisfying activities. One should consider which activities are best supported by IT, taking into consideration job satisfaction in addition to automation. Socio-technical design is argued for and it is maintained that IT and social systems evolve together, each adapting to the other (referring to Eason and Mumford). Technologies ‘can transform tasks in exciting and satisfying ways, BUT incremental changes to existing practice are easier to understand and adopt’ (p. 82). Cooperative design is also argued for and an emergent view can be identified, in that it is underlined that use always evolves, activities co-evolve with the technology in use and technologies have both intended and unintended consequences.

SBD strongly advocates user involvement. Cooperative design is recommended, and mutual learning by developers and users is emphasized. Problem analysis entails the assessment of a problem situation by means of interviews and field studies. Brainstorming among users and developers is also mentioned as is the participatory analysis of videotapes of work activities. Activity and information design scenarios can be refined and prototype and mock-up experiments are conducted in cooperation with the users. Empirical usability evaluation is also included. Finally, surrogate user involvement is also advocated. Specific activities are assigned to usability engineers in the implementation of the method, and it is also suggested that these people should carry out analytical usability evaluations that do not involve real users.

Power and politics are addressed rather weakly in SBD. Power and politics inside the development team/organization are mentioned, and it is noted that usability engineers should acknowledge that trade-offs are always needed. PD is advocated, but power and politics among the user population or in the design process are not mentioned. The method acknowledges that safety, access, the digital divide and people with special needs have to be acknowledged in addition to usability, and ethical usability is to be striven for.

SBD recognizes personalization as an issue and discusses it in length from the viewpoint of adaptivity. It is mentioned that custom interfaces may be developed for user groups and that the information design may differ for different users. Complex models of human behaviour can be built as an aid to task performance and the prediction of preferences and decisions (p. 304). It is also mentioned, however, that there are risks with software that is intelligent, although in spite of this, there is a whole section in the book devoted to intelligent interfaces (pp. 317–325).

The HCSDLC Method

HCSDLC (Zhang et al., 2005; Te’eni et al., 2007) is a systems development method that integrates HCI development into the systems development lifecycle process. The method is composed of five phases: project selection, planning, interaction analysis, design and implementation. It claims to be a truly human-centred IS development method (Zhang et al., 2005) that considers both organizational and human needs (Te’eni et al., 2007, p. 293). According to Zhang et al. (2005), ‘human-centred systems development’ covers both basic user-centred
systems functionalities and human-centred human–computer interaction development (p. 513). Te’eni et al. (2007) clarified ‘user-centred’ systems development as meaning that one starts the design process with an understanding of the user and the user’s tasks (p. 227).

HCSDL has a strong user focus. One of the first steps in the interaction analysis phase is user analysis, which identifies the target users and their characteristics in terms of demographic data, traits and intelligence and of job- or task-related factors (Zhang et al., 2005, p. 526). Overall, HCSDL has a strong human factor (human constraints, limitations, needs, strengths and potentials) orientation (Zhang et al., 2005, p. 520, 522) as also evidenced by the book of Te’eni et al. (2007), which covered the physical, cognitive and affective sides of humans (chapters 4–6). These human factors can be used to understand individual users and identify typical users.

At the same time, HCSDL has a strong work orientation. The interaction analysis includes task analysis as one of its major steps and distinguishes organizational-level tasks and tool-level tasks. The analysis starts with the organizational-level tasks, and after decomposing them, ends up with tool-level tasks involving how to interact with the system or its tools. Zhang et al. (2005) noted, however, that ‘a key issue in building new systems is to realize that the new systems change skill sets and obstruct current workflow. Development of a new system must take into account the movement from one type of structured work environment to another’ (p. 526). In this respect, the method assumes the change to be technology-driven. Yet, the main emphasis is on supporting current work processes by means of technology rather than on reconsidering and redesigning the organizational-level tasks. This observation is also supported by the way in which Te’eni et al. (2007) introduced the CD method (Beyer & Holtzblatt, 1998). They noted that it includes seven parts, but they introduced only the first three (contextual inquiry, work modelling, consolidation) and did not discuss work redesigning in their book. The method is strongly based on the idea of a fit between the human user, the tasks to be performed and the technologies available in a given context. A good fit is assumed to lead to better performance and well-being (Te’eni et al., 2007, pp. 7–8). It seems, however, that Te’eni et al. (2007) saw technology as a major source of change. A change in this component may disturb the fit, implying that it may be necessary to change the tasks and to develop new skills in order to restore the fit, but the change does not start from the tasks or the human participants. Te’eni et al. (2007) also included weak signals pointing to the idea of emergence: ‘Finally, the workplace, with its computers and other artifacts, does not always behave as expected. People behave in unpredicted and unplanned ways’ (p. 229).

To some extent, HCSDL covers both holistic work models and local work practices. There is an implicit link between CD and HCSDL in Te’eni et al. (2007) as Te’eni et al. referred to the working habits of workers. These can be interpreted as implying local work practices, but the question of how to learn to know these is not specifically addressed in Te’eni et al. (2007). In the case of holistic work models, Te’eni et al. emphasized that ‘managing interrelated tasks, even if not performed concurrently, requires the user to maintain a global view of the tasks, understand thoroughly their interrelations and how changes in one task affect the other’ (p. 233). They also referred to several frameworks familiar from the IS literature, such as Anthony’s model (p. 345), the classification of tasks into structured, semi-structured and unstructured
ones (p. 22), and the classification of workers into office support staff, knowledge workers, middle managers and executives (p. 23).

HCSDLC does not specifically address the question of user involvement. Zhang et al. (2005) totally omitted it, and Te’eni et al. (2007) stated ‘The thrust of user-centered systems development is that users of an information system should play an important role during the systems development process, especially at the early stages’ (p. 294). More specifically, this means ‘involving users as much as possible so that they can influence design’ (p. 294). They also noted the social responsibility of each IS developer to ask ‘Have I arranged for adequate participation of users in the design process?’ (p. 361). They referred to PD but did not discuss user involvement in any more detail.

The interaction analysis in HCSDLC includes context analysis, which implies understanding the technical, environmental and social settings in which the systems will be used (Zhang et al., 2005). Interestingly, it does not include any explicit references to power or politics, although the general statements about interactions with other organizational entities and organizational policies and practices may be interpreted as referring to power relationships. Zhang et al. (2005) did emphasize, however, the need for users to control their work tasks (pp. 513, 517). In the context of systems having external users, they remark that the ‘significance of organizational context may be less than that in the internal use situation and the role of such organizational context may be less controllable by the system developers’ (p. 526), which primarily refers to power in the hands of systems developers vis-à-vis external users. Te’eni et al. (2007) were also quite silent about power issues, discussing it most explicitly in the context of ERP systems, where they note that ‘Although an ERP tightens management control by monitoring almost all core organizational activities, another outcome of an ERP is to empower employees through process design. Management often resists empowerment by adding new process constraints on top of the RTP processes’ (p. 44). They also advocated user empowerment in the sense of providing the right tool set that ‘empowers the user with the capabilities to solve creatively new situations and problems unforeseen by the designer’ (pp. 219–220) and designing ‘systems that support the internal locus of control’ among users (p. 365). Te’eni et al. (2007) also mentioned power games as group losses to be minimized in the context of Group DDS (p. 335) and power distance as a dimension of national culture (p. 339). Furthermore, power is also included in voluntariness of use and the discretion of users (pp. 23, 32), low-status members in the context of Group Support Systems (p. 39).

Zhang et al. (2005) did not discuss the issue of personalization at all, but Te’eni et al. (2007) identified the question ‘Have I considered individual differences among users in the design of my system?’ as one aspect of the social responsibility of each IS developer (p. 361) and identified support for a diversity of users as one of their design guidelines (p. 202). Despite this recognition, they did not discuss how to make the system more adaptive or adaptable beyond quite general remarks that ‘The system must therefore be designed for flexibility (. . .) Additionally, individual styles of thinking and feeling will play a greater role in unstructured compared with structured tasks. Being able to adapt to personal tastes should therefore be another design concern’ (p. 233).
4.2. Summary of the findings

The results of the analysis are summarized in Table 3, which demonstrates that although all four methods are introduced as ‘user-centred’, they differ radically from each other in the way they address the four dimensions of user-centredness. Apart from CD, they all have a strong user focus, but GDID and SBD focus mainly on fictive users (personas or actors), although they also recognize human factors. HCSDLC is mainly concerned with human factors.

All the methods analysed have a strong work focus, but they differ considerably at a more detailed level. GDID pays attention to work redesigning rather than current work practices, while the focus of HCSDLC lies on the current work practices rather than work redesigning. CD and SBD cover both. The main focus of GDID and SBD is on local work practices, whereas HCSDLC is just the opposite in this respect – its emphasis is on holistic work models. Again CD covers both. In the case of change, we interpret HCSDLC as viewing this mainly as technology-driven, while GDID sees it mainly as interactive. CD and SBD seem to be richer in this respect, for while CD mainly sees change as work-driven and SBD as interactive, they both also recognize interactive, work-driven and emergent aspects of change.

The four methods also differ in the case of user involvement. HCSDLC is largely silent about the issue, SBD places a strong emphasis on user involvement and its attitude towards this is generally positive, while CD includes some user involvement, but the authors, Beyer and Holtzblatt, acknowledge that when following the CD method, it may be that users are contacted only in the first and last phases of the design process so that user involvement does not play a very significant role. GDID includes quite limited user involvement, and its attitude towards direct user involvement is negative. CD, GDID and SBD argue for surrogate user involvement in the sense that it is assumed that there should be a CD team, interaction designers or usability engineers acting as user advocates in the development process. Overall, the four methods are weak in the extent to which they address power issues related to systems development, and HCSDLC largely omits this aspect.

In the case of system personalization, CD and HSDLC recognize the issue but do not really address it. SDB and GDID clearly pay more attention to this opportunity.

5. DISCUSSION AND CONCLUSIONS

As noted in the introduction, UCD has recently gained increasing attention. At the same time, it has remained as a concept somewhat loosely defined. Using the existing literature on UCD, four dimensions of user-centredness were recognized: user-centredness as a focus on the user, user-centredness as work-centredness, user-centredness as user involvement and user-centredness as system personalization. We also identified a variety of options in the way these dimensions are addressed in the UCD literature. The conceptual variation in user-centredness is summarized in Table 2.

To demonstrate the usefulness of the proposed framework, we applied it to the analysis and comparison of four systems development methods that are claimed to be user-centred: GDID,
Table 3. Summary of user-centredness in four systems development methods

<table>
<thead>
<tr>
<th>GDID</th>
<th>CD</th>
<th>SBD</th>
<th>HCS DLC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User focus:</strong></td>
<td><strong>Strong user focus:</strong></td>
<td><strong>Some user focus:</strong></td>
<td><strong>Strong user focus:</strong></td>
</tr>
<tr>
<td>how can one identify and represent users?</td>
<td>main focus on fictive users ( personas), constructed on the basis of empirical data (resembling user profiles)</td>
<td>individual users interviewed and observed, individual work models produced, but the main focus is on work, not on the user as a human being</td>
<td>fictive users (actors) constructed on the basis of empirical data</td>
</tr>
<tr>
<td>some focus on human factors</td>
<td>some focus on human factors</td>
<td>user profiles produced</td>
<td>user profiles produced</td>
</tr>
<tr>
<td><strong>Strong work focus:</strong></td>
<td><strong>Strong work focus:</strong></td>
<td><strong>Strong work focus:</strong></td>
<td><strong>Strong work focus:</strong></td>
</tr>
<tr>
<td>covers current and future practices; emphasis on the latter</td>
<td>covers current and future practices; covers holistic work models and local work practices</td>
<td>covers current and future practices; covers holistic work models and local work practices</td>
<td>covers current and future practices; covers holistic work models and local work practices</td>
</tr>
<tr>
<td>covers holistic work models and local work practices; emphasis on the latter</td>
<td>change primarily work-driven; interactive and emergent views discussed</td>
<td>change primarily work-driven; interactive and emergent views discussed</td>
<td>change primarily interactive; work-driven and emergent views discussed</td>
</tr>
<tr>
<td><strong>User involvement:</strong></td>
<td><strong>Limited user involvement:</strong></td>
<td><strong>Some user involvement:</strong></td>
<td><strong>Strong user involvement:</strong></td>
</tr>
<tr>
<td>why should users be involved?</td>
<td>users interviewed and observed during contextual inquiries, invited as co-interpreters</td>
<td>users interviewed and observed during paper prototyping sessions, also invited as co-designers and co-interpreters</td>
<td>users interviewed and observed during field studies</td>
</tr>
<tr>
<td>how should users be involved?</td>
<td>users interviewed and observed; are not allowed to affect the design solutions</td>
<td>surrogate user involvement: contextual design team as a user advocate</td>
<td>users interviewed and observed during usability evaluation</td>
</tr>
<tr>
<td>who has the power to decide about change?</td>
<td>surrogate user involvement: interaction designers as user advocates</td>
<td>power questions addressed to some extent; cultural issues and power relations modelled; more power demanded for interaction designers, users given no power</td>
<td>users may also participate in field study analysis, brainstorming, work redesigning, development of prototypes</td>
</tr>
<tr>
<td></td>
<td>power questions weakly addressed, some cultural and ethical issues discussed, more power demanded for interaction designers, users given no power</td>
<td></td>
<td>users may be assigned some power of decision</td>
</tr>
<tr>
<td>System personalization</td>
<td>Discussed from the viewpoint of adaptivity and adaptability</td>
<td>Recognized but not really addressed</td>
<td>Discussed especially from the viewpoint of adaptivity</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Recognized but not really addressed</td>
</tr>
</tbody>
</table>

GDID, Goal-Directed Interaction Design; CD, Contextual Design; SBD, Scenario-Based Design; HCS DLC, Human-Centred Systems Development Life Cycle.
CD, SBD and HCSDLC. The analysis showed considerable variation in how these methods address the four dimensions of user-centredness. The findings of the comparative analysis were summarized in Table 3 earlier.

The analysis of the four user-centred methods points to the major limitation of the present paper. The findings are based on our interpretation of the material that we had available. It may be that we were not able to capture all the intentions and ideas of the developers of the methods in question. On the other hand, the possible misunderstandings are indicators that those points in the documentation of the methods could be clarified. Another limitation is that our analyses relied solely on books that described the methods. It was shown decades ago that there may be clear differences between how methods are used in practice and how they are described in books (see Hirschheim & Newman, 1991; Nandhakumar & Jones, 1997). It could be an interesting topic for another study to examine how user-centredness is realized in practice when these allegedly user-centred methods are in use.

Acknowledging these limitations, we believe that a deeper understanding of the concept of user-centred(ness) is useful for both researchers and practitioners. The framework helps researchers and practitioners make sense of the growing body of literature on UCD, while practitioners interested in making their development process more user-centred can evaluate the methods they use according to the framework or use it while selecting among multiple allegedly user-centred methods. The methodological analysis carried out here may be useful for them in that it indicates in what sense these methods are user-centred and what limitations they may have as UCD methods. The dimension of user-centredness and its related options also provide an analytical framework for future research into the topic. One specific topic would be to investigate possible incompatibilities between different ideas and options involved in user-centredness (see section 3.5).

One should also note that each of the four dimensions of user-centredness has its own research challenges. Related to the user focus, the present paper identified alternative means of identifying and representing users. To our knowledge, there is not much research on how effective these are and in what contexts. As pointed out in section 3.2, the focus of UCD turned from a user focus to a work-centred one during the 1990s. Despite this change, we feel that there are many research opportunities related to the user as a human being as evidenced by the current interest in user experience (e.g. McCarthy & Wright, 2004). This interest is partly explained by the fact that many recent IT applications are mainly used outside the workplace context (cf. Wright & McCarthy, 2005). It seems that this is pushing the pendulum back from work-centredness to a focus on the user.

Norman (2005) suggested an activity-centred approach where the term ‘activity’ may be interpreted as covering activities such as playing computer games and spending time and fantasizing in the virtual world. As Norman (2005) pointed out that technology often defines the activity in these cases – it does not serve or support the activity as is normally assumed but may, in effect, constitute the activity. The extension of work-centredness to activity-centredness is clearly a research challenge for the future.

While Markus & Mao (2004) found user involvement to be an ‘old, tired concept’, they also noted that it has been mainly discussed in the context of systems analysis and design but not
so much in the context of systems implementation. This is naturally the situation in the UCD literature. Although the implementation phase is critical in all systems development (Swanson, 1988), application packages have increased its prominence: in their case, all the problems of systems development are encountered during the implementation phase (Iivari, 1990). One should note that the implementation of ERP-based IS, for example, includes a variety of development tasks such as the redesigning of business processes and organizational structures, and also change management, i.e. preparing the user population to adopt the system, in addition to the conventional implementation activities such as testing, installation or conversion of the system, training, and evaluation of its performance. Actually, a customer-centred ERP implementation method has been developed that relies on the CD method (Vilpola 2009), but with the exception of that paper, the UCD literature remains quite silent about user-centred implementation process.

Personalization is a research area of its own that is now under relatively active research. It includes several issues such as what characteristics of users it is relevant to model, what features of systems can be meaningfully personalized, what is the mapping between user characteristics and personalized system features, how a user model can be created, and what is the value of personalization to the users and other stakeholders. Personalization, especially adaptivity, if totally hidden to the user, also includes ethical and privacy problems (Kobsa, 2002). There is also a clear need for more empirical studies on adaptable and adaptive systems. Albert et al. (2003), for example, reported that users are unenthusiastic towards adaptive systems that attempt to infer user needs, goals and interests, essentially suggesting that at least not all ideas of personalization are universally valued by users.

The four dimensions of user-centredness may also guide the creation of more user-centred systems development methods. The dimensions lead us to view user-centredness more as a quantity than a quality. Rather than asking which systems development methods are user-centred and which are not, it is more appropriate to question to what extent and in what sense different systems development methods are user-centred. The options related to each dimension provide a framework for reflecting upon the sense in which a systems development method may be made user-centred. The four user-centred methods analysed in section 4 illustrate alternative strategies for developing user-centred methods. GDID, CD and SBD were developed as user-centred methods without explicitly integrating them into any existing methods, while the Human-Centred Software Development Life Cycle was deliberately developed as an integrated approach. Gulliksen et al. (2003) described the use of UCD in connection with the Rational Unified Process, reporting considerable problems. Given the variety of user-centredness, there is a clear need for additional research into the problems of integrating ideas of user-centredness with existing systems development methods.

**ACKNOWLEDGEMENT**

We are grateful to Professor Ping Zhang for her comments on an earlier version of the present paper.
REFERENCES


Cooper, A. (1999) The Inmates Are Running the Asylum: Why High-Tech Products Drive is Crazy and How to Restore Sanity. Macmillan, Indianapolis, IN, USA.


© 2010 Blackwell Publishing Ltd, Information Systems Journal 21, 125–153
nition of usability: analyzing ISO 13407 against ISO 9241-
theory and information systems research. MIS Qua-
Karat, J. (1997) Evolving the scope of user-centered
Karat, C.M., Brodies, C., Karat, J., Vergo, J. & Alpert, S.R.
centered focus in human–computer interaction field. IBM
Systems: An Organizational Perspective. Addison-
Wesley, Reading, MA, USA.
Engineering Informatics, 23, 142–148.
issues and concerns. Computer Supported Cooperative
Kinsch, L.J. & Beath, C.M. (1996) The enactments and
consequences of token, shared, and compliant partici-
pation in information systems development. Accounting,
Management and Information Technologies, 6, 221–
254.
Klein, G., Kaempf, G.L., Wolf, S., Thorsden, M. & Miller, T.
(1997) Applying decision requirements to user-centered
design. International Journal of Human-Computer
Studies, 46, 1–15.
ing user-centered criteria for information systems: a
software ergonomics perspective. Information & Man-
agement, 11, 163–172.
Kobsa, A. (2002) Personalized hypermedia and interna-
hypermedia presentation techniques for improving
online customer relationships. Knowledge Engineering
Review, 16, 111–155.
Kramer, J., Noronha, S. & Vergo, J. (2000) A user-
centered design approach to personalization. Communi-
Kujala, S. (2003) User involvement: a review of the ben-
efits and challenges. Behaviour & Information Technol-
ogy, 22, 1–16.
development. In: The Proceedings of the Conference on
Human Factors in Computing Systems, Adelson, B.,
Dumais, S. & Olson, J. (eds), pp. 3–9. ACM, New York,
USA.
Liang, T. & Tanniru, M. (2006–2007) Special Section: cus-
tomer-centric information systems. Journal of Man-
Prentice-Hall, Englewood Cliffs, NJ, USA.
Systems CH’90, pp. 175–182. Seattle, WA.
Markus, L. (1983) Power, Politics, and MIS Implementa-
theory for systems that support emergent knowledge
processes. MIS Quarterly, 26, 179–212.
Markus, M.L. & Mao, Y. (2004) Participation in develop-
ment and implementation – updating an old, tired
concept for today’s IS contexts. Journal of the Associa-
tion for Information Systems, 5, 514–544.
and organizational change: causal structure in theory
and research. Management Science, 34, 583–598.
Mayhew, D.J. (1999) The Usability Engineering Lifecyle: A
Practitioner’s Handbook for User Interface Design.
Morgan Kaufmann Publishers, Inc, San Francisco, CA,
USA.
McCarty, J. & Wright, P. (2004) Technology as experi-
ence. Interactions, 11, 42–43.
Technology. The ETHICS Method. Manchester Busi-
ness School, Manchester, UK.
dark: the changing user–developer relationship in infor-
mation systems development. Proceedings of 18th Inter-
Boston, MA, USA.
participative discourse amongst IT operations person-
nel. Proceedings of 10th Australasian Conference on
Information Systems, pp. 691–702.
Centered Design: New Perspectives in Human–
Computer Interaction. Norman, D.A. & Draper, S.W.
(eds), pp. 31–61. Lawrence Erlbaum Associates, Hills-
dale, NJ, USA.


Biographies

**Juhani Iivari** is a Professor in Information Systems at the University of Oulu, Finland. He received his PhD degrees from the University of Oulu in 1983. He is a Senior Editor of *MIS Quarterly* and serves in editorial boards of four other IS journals. His research has broadly focused on theoretical foundations of information systems, design science research in information systems, information systems development methodologies and approaches, organizational analysis, implementation and acceptance of information systems and the quality of information systems. He has published in a number of journals such as *Communications of the ACM, Data Base, European Journal of Information Systems, Information & Management, Information and Software Technology, Information Systems Journal, Information Systems Research, Journal of the Association of Information Systems, Journal of Management Information Systems, Journal of Organizational Computing and Electronic Commerce, MIS Quarterly, Omega, Scandinavian Journal of Information Systems*, and in numerous conference proceedings.

**Netta Iivari** received her doctoral degree in Information Systems from University of Oulu in 2006. She also has a master’s degree in Cultural Anthropology from University of Jyväskylä. She currently holds the position of a Post-Doctoral Research Fellow from the Academy of Finland. She was appointed as an Adjunct Professor in the Department of Information Processing Science, University of Oulu in 2009 with the field ‘user-centred, participative information systems design’. Her research is strongly influenced by interpretive and critical research traditions. Her long-lasting research interests are related to the empirical, interpretive examinations of participation of different kinds of stakeholder groups in defining, developing, shaping and framing information systems. Her empirical investigations have mainly been carried out in new IS contexts: in packaged software and open source software development contexts and related to solutions targeted at children. Currently, she is specifically interested in participatory design with children and in user participation in open source software development. She has published related to these matters in several IS conferences and journals.