Global Co-Operation in the New Millennium
The 9th European Conference on Information Systems
Bled, Slovenia, June 27-29, 2001

TOWARDS MORE PROFESSIONAL INFORMATION SYSTEMS DEVELOPMENT: ISD AS KNOWLEDGE WORK

Juhani Iivari

Department of Information Processing Science, University of Oulu, 90014 Oulun yliopisto, Finland
Tel.: +358 8 5531922, Fax: +358 8 5531890.
juhani.iivari@oulu.fi

Rudy Hirschheim

Bauer College of Business, University of Houston, Houston, TX 77204-6282, USA
Tel.: 713 743 4692, Fax: 713 743 4693
rudy@uh.edu

Heinz K. Klein

Department of MIS , Fox School of Business and Management, Temple University, Philadelphia, PA 19122-6083, USA
Tel.: 215 204 3059, Fax: 775 855 2843
klein@acm.org

ABSTRACT

The purpose of this paper is to argue that research and practice of ISD could substantially benefit from interpreting ISD as knowledge work. Two distinct areas of competence are identified for system developers: IS application knowledge and ISD process knowledge. Focusing on process knowledge, the authors propose a template for organizing the body of process knowledge in ISD consisting of four levels: systems development techniques, methods, approaches and paradigms. The conclusions discuss implications of the knowledge work perspective for the education of IS experts

1. INTRODUCTION

A recent joint project of the ACM and IEEE Computer Society proposed a guide for the software engineering (SE) body of knowledge (SWEBOK 2000) as well as a code of ethics and professional practice (SWECOE 2000). This project leads directly to the question of *whether the IS field should follow their lead and take similar action to establish IS as a profession*. Although one goal of professionalization is to enhance the knowledge and expertise of the members of the profession, it also involves a political agenda. McConnell and Tripp (1999) report that some states in the USA, and parts of Canada, have started the licensing of professional software engineers. The same is happening in the U.K

Steering ISD in a professional direction leads to a second question: *what are the distinctive competence area(s) of IS experts*? We suggest two areas, which will be elaborated in sections 2 to 4: IS application knowledge and ISD process knowledge. An analysis of the ten knowledge areas listed in SWEBOK (2000) shows that they do not include any knowledge about applications. The SE process, on the other hand, is extensively addressed. Our conclusion is that IS experts have a distinctive competence of aligning IT artifacts (IS applications and other software products) with the organizational and social context in which the artifact is to be used, and in the organizational implementation and evaluation of these artifacts and related changes. Further, the IS expert aligns these artifacts with the needs of the people who are supposed to use the system. A rudimentary comparison of this with the ten knowledge areas in SWEBOK (2000) is sufficient to show that this competence is virtually ignored or at best weakly taken into account in SWEBOK.

Section 3 discusses the distinctive competencies of IS experts, and especially that of aligning IT artifacts with the organizational and social context in which the artifact is to be used and to the needs of people who are supposed to use the system. Much of this competence is based on research into IS. We claim, however, that it is not sufficient from the viewpoint of practitioners that such knowledge is reported as 'practical implications' of research results in countless number of research articles, but rather this knowledge should be condensed and packaged in an action-oriented way so that IS practitioners find it helpful in their work. We suggest that ISD approaches and methods are appropriate places to insert this action-oriented knowledge. Section 3 suggests how this can be accomplished by a rich knowledge representation scheme for ISD process knowledge.

One of the major research findings into ISD method use is that methods have not been very successful in guiding practice. They are not used extensively, and as far as they are used, they are not literally applied (Hardy *et al*. 1995; Wynekoop and Russo 1993). Even though existing research on the actual use of ISD methods can be criticized on a number of grounds, it clearly indicates that the relationship between ISD methods and ISD practice needs careful attention. Section 4 discusses this relationship emphasizing that ISD practice is not a uniform phenomenon but differs considerably in the extent to which it can be made "professional". We also claim that the application of ISD approaches and methods comprises a parallel instantiation process. Generally, this instantiation is a continuous process that takes place in close concert with the ISD practice as it unfolds.

2. PROFESSIONS AND THE CONCEPT OF KNOWLEDGE WORK

2.1. Professions and Professionalization

Professions and professionalization are widely discussed in sociology (Macdonald, 1995) and organization theory (Minzberg, 1983). There have also been a few attempts to analyze the IT occupation as a possible profession (Ford and Gibbs 1996; Orlikowski and Baroudi 1989; Zwerman 1999). Professions are often interpreted in terms of a number of criteria such as a service ideal, professional culture and associations, accredited education, code of ethics, etc. Based on criteria such as these Ford and Gibbs (1996) conclude that SE as an occupation does not fulfill the traits of a profession. Freidson (1988) strongly argues that the only valid criterion for distinguishing professions is their autonomy, that is their right, given officially by the state and implicitly by the public, to control their work. Following Freidson (1986), Orlikowski and Baroudi (1989) claim that IS workers (including operators, programmers, analysts and various technical specialists) cannot be considered professions. Zwerman (1999) identifies an ahistorical orientation of software developers as one obstacle to the professionalization of software development.

It is widely accepted that professionalization efforts have an ideological and political aspect of increasing the status of the occupation in question, backing its 'exclusionary shelters in the market' or providing a market monopoly (Freidson 1986,1988; Macdonald, 1995). Abbott (1988) provides a relatively recent account of professionalization, interpreting it as competition between different occupational groups for jurisdiction. In distinction to these contributions, our intention is not to discuss the political side of professionalization of the IT occupation nor to advocate professionalization. Instead we will focus on analyzing ISD as knowledge work, suggesting that this is a more realistic avenue to enhance the expertise of IS developers and thereby their "professionalism". The perspective of KW leads one to focus on the BoK of systems developers and its application in ISD. It points to the core of professional work without getting involved in the political battle of professionalization. Zwerman (1999, p. 66), for example, identifies the "exclusive command of systematic esoteric body of knowledge, as the single most important defining characteristic of professionals", and Abbott (1988) notes "Despite their substantive differences, (…) all agreed that a profession was an occupational group with some special skill. Usually this was an abstract skill, one that required extensive training. It was not applied in a purely routine fashion, but required revised application case by case" (p. 7).

2.2. Knowledge Work

Knowledge work (KW) is difficult to define precisely because all work requires knowledge to some extent (Beyerlein *et al*. 1995). As an initial characterization, we propose four criteria to distinguish KW from other work: (i) KW is based on a BoK, (ii) entails working on representations (data) of the objects of work, (iii) stipulates a deep, theoretical understanding of the objects of work, and (iv) KW produces results, which entail knowledge as their essential ingredient (Iivari and Linger 1999). This characterization emphasizes the significance of a BoK in undertaking KW. It also highlights the abstract and detached nature of KW since a knowledge worker often works indirectly through the representation of the object of work. KW requires intellective skills (Zuboff 1988), in contrast to action-centered skills, KW is very abstract, because the worker must understand the symbolic reference relationships between the objects of work and their symbolic representations (Zuboff 1988). In ISD, the objects of work - information and software systems - are the reasons why ISD is considered to be so difficult. Brooks (1987), for example, posits the conceptual nature, complexity, conformity to possibly idiosyncratic interfaces, expected changeability, and the inherent invisibility of software as essences of software systems which make software development so different from other engineering disciplines.

The first and the last characteristics above imply that we interpret KW primarily as knowledge-applying (in the sense of applying the BoK) rather than knowledge-producing work. This is in contrast to many interpretations of KW, or closely related concepts, as primarily knowledge producing work (Machlup 1962; Schultze 2000). Our interpretation is based on a number of points. Firstly, the Maclupian tradition leads to a number of artificial demarcation problems. Secondly, especially when interpreting knowledge as a stock (Schultze 2000), one should be more specific with regard to knowledge to whom - the individual professional, possible employer, to the paying customer, or somebody else? Thirdly, referring to Schultze's interpretation of KW as covering professionals, we claim that a customer of a professional (in medicine and law, for example) often interprets that (s)he is paying for a service rather than for gaining knowledge, and often it is not necessary for a customer to understand the exact knowledge content of the output. Let us take as an example a software engineer who adapts a program to a new platform. Note, however, that our interpretation does not deny that knowledge workers and even their employer organizations are learning something through their work. In that sense, KW is producing knowledge. The point is that the knowledge production aspect is not primary, and hence not very distinctive to KW. Moreover, our interpretation of the domain of KW does not exclude knowledge-producing work because typically knowledge-producing work (such as research and development) is also knowledge-applying work. The knowledge applying nature of KW will be now elaborated in section 2.3.

2.3. Body of Knowledge

The body of knowledge (BoK) is knowledge of the relevant phenomena associated with KW as an activity. It comprises facts, rules, techniques, case histories (cases), stories, theories, hypotheses, philosophies, metaphors, etc. The BoK is not necessarily constrained to the codified knowledge generally accepted as valid and taught in educational institutions. Indeed, Abbott (1988) argues that "academic professional knowledge" is more symbolic than practical (p. 54). On the other hand he does not deny that that the practical diagnostic and treatment classification systems are based on "academic professional knowledge". Actually, his point is that they are organized differently. The theoretical classification system is organized along logically consistent, rationally conceptualized dimensions, while the practical diagnostic classification system, for example, is probabilistic, proceeding from common to esoteric (p. 42). Nor is the BoK required to be scientifically valid (see Feltovich *et al*. 1989); it may include experiential knowledge. According to Freidson (1988), experiential knowledge is extremely significant in established professions such as medicine.

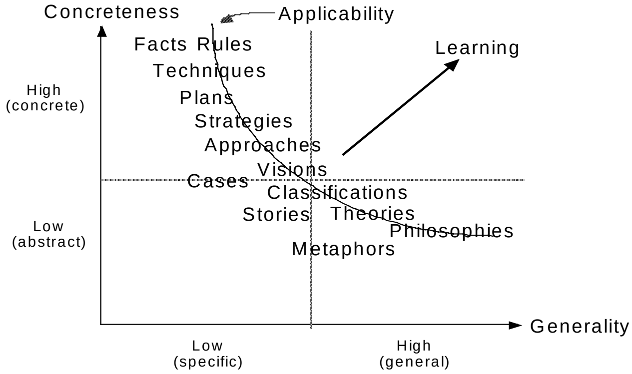

Figure 1: *The concreteness and generality of the body of knowledge*

Our focus here lies in the relationship between the BoK and the objects of work rather than on the abstract concept of the BoK isolated from its application in KW. This relationship may be referred to as 'applicability' and can be interpreted as a function of the concreteness and generality of the BoK (Figure 1). *Concreteness* (C) describes how directly the BoK can be applied in the work. When concreteness is low, the application requires the BoK to be made more concrete based on judgement and experience. *Generality* (G) describes the range of different cases and situations covered by the BoK. When generality is low, the application of the BoK to a situation outside the scope of the BoK requires generalization based on discretionary judgement.

Figure 1 illustrates the trade-off between generalizability and concreteness in that it is very difficult to be very general without losing concreteness and vice versa. If we define the *applicability* of the BoK as a function of its concreteness and generality, the first derivatives of which are positive, the trade-off curve in Figure 1 can be interpreted as an applicability frontier. At the same time Figure 1 suggests, that we may move the applicability frontier towards higher generality and concreteness through learning. Learning may imply better organization of the BoK. This organization could be a better understanding of the relationship between general knowledge (theories) and concrete knowledge (facts), or it may be an outcome of scientific progress (more general theories). For example, cases can be interpreted as comprising facts (events) and a story providing at least one plausible explanation for the interrelationship between the facts.

So, if we apply the flow-stock metaphor of information and knowledge (Nonaka and Takeuchi 1995; Schultze 2000) our point is the BoK is not just a stock but an organized stock. Abbott (1988) underlines the significance of abstract knowledge as a source from which practical skills grow, and for the survivability of a profession. In fact, he claims that “For me this characteristic of abstraction is the one that best identifies a profession” (p. 8). In view of the rapid technological development in the IT field we also wish to stress the significance of abstract, general knowledge in the BoK of IS developers. At the same time Abott (1988) emphasizes the need for “an equilibrium between an extreme abstractness and extreme concreteness” (p. 104). The practical nature of KW requires that the actor has concrete knowledge. On the other hand, it is essential that in general, (s)he also has abstract knowledge. The following section will elaborate the body of ISD process knowledge in more detail.

3. ISD PROCESS KNOWLEDGE

3.1. Knowledge Areas in IS Development

Applying Freeman (1985) one can distinguish three types of knowledge involved in ISD: Technology knowledge, application domain knowledge, and systems development process knowledge. Jones and Walsham (1992) propose organizational knowledge as an additional category, distinct from the application domain knowledge. They relate organizational knowledge to knowledge “about the social and economic processes in the organizational contexts in which the IS is to be developed and used”. We would also like to include more explicitly the work processes in the organizational context to be supported by the IS. In addition to the four knowledge types, we suggest a fifth category, application knowledge. Application knowledge is knowledge about typical applications, their structure, functionality, behavior and use, in a given application domain, and knowledge of possibilities to support the application domain using IT. The knowledge of possibilities to support the application domain using IT is a creative synthesis of the application domain knowledge, organizational knowledge, and technical knowledge. To sum up, we posit five knowledge areas: technical knowledge, application domain knowledge, organizational knowledge, application knowledge, and ISD process knowledge. They also correspond to four of the six knowledge areas identified by Vitalari (1985): Our application domain knowledge to his “functional domain knowledge”, our application knowledge to his “application domain knowledge”, our organizational knowledge to his “organizational specific knowledge”, and our ISD process knowledge to his “knowledge of methods and techniques”.

It is now appropriate to return to the question: Where may the distinctive knowledge of IS developers lie? It cannot be in the technical knowledge where experts in Computer Science and Software Engineering are likely stronger. Neither can it be in the application domain knowledge where people working in the application domain are likely more knowledgeable. It is the same situation with the organizational knowledge. The only remaining candidates are application knowledge and ISD process knowledge. Even though we suggest the two as distinctive knowledge areas of IS experts, we do not attempt to deny the significance of the remaining three in ISD. On the contrary, there can be no doubt about the importance of technical knowledge, and there is also ample evidence about the significance of application domain knowledge (Curtis *et al*. 1988) and organizational knowledge (Vitalari 1985).

In the following we will focus only on ISD process knowledge. Referring to ISD process knowledge, we claimed earlier that the distinctive competence of IS experts lies in their expertise of aligning IT artifacts with the organizational and social context in which the artifact is to be used and to the needs of people who are supposed to use the system, in the organizational implementation and evaluation of these artifacts and related changes. We are speaking about IT artifacts instead of traditional ISs because many software artifacts, which are not considered traditional ISs, comprise more and more features which resemble ISs. Consider for example, an embedded computer system in a mobile telephone. As the functionality of these systems expands, the embedded software does not only implement some of the necessary functions of the telephone in contrast to hardware implementation, but the software provides a number of auxiliary services to users of the mobile phone. The meaningfulness of these auxiliary services can be assessed only against users’ needs.

One can identify four competency areas or domains in the above interpretation of core competence: Firstly, organizational alignment of IT; secondly, user requirements construction (engineering, analysis, elicitation and specification); thirdly, organizational implementation; fourthly, evaluation/assessment of IT artifacts. We make a distinction between the first two because organizational alignment and user requirements construction may be quite distinct activities, for example in an ISD project involving business process redesign or reengineering.

Alignment of IT plans and organizational objectives has consistently been reported among the key concerns of IT managers and business executives (e.g. Brancheau *et al*. 1996), despite the concerns of Reich and Benbasat (2000) who suggest there is no comprehensive model of this “alignment” construct. It is beyond the scope of this paper to discuss the concept in detail. Our only point is that we interpret alignment to reflect the 'fit' of an IT artifact, an information system or a software product, with the organizational and social context of its use rather than the 'fit' between IT plans and organizational objectives. This latter alignment may support our notion of alignment, but it is not necessary for it to be based on formal IT planning.

Requirements construction continues to be the major bottleneck in ISD (Kumar and Welke 1992). It is, of course, one of the knowledge areas in SWEBOK (2000). Our contention is, however, that user requirements construction *together with* organizational alignment form the core competence of IS experts. It is beyond the scope of this paper to explore how requirements construction is understood in various communities. However, one can observe that the problem of organizational alignment is largely neglected in the SE tradition. We also wish to emphasize the richness of approaches to requirements construction identifiable in the IS community (Iivari and Hirschheim 1996).

The third domain - organizational implementation - refers to the implementation research tradition in IS.[1] The problem of organizational implementation is totally neglected in SWEBOK despite the fact that organizational implementation is often problematic (Swanson 1988). In the case of the fourth domain, evaluation/assessment of IT artifacts, this too is largely ignored in SWEBOK. Yet again this is an area richly explored by the IS field (DeLone and McLean 1992; Smithson and Hirschheim 1998).

There is a rich body of literature within IS that addresses the issue of making an information system 'fit' its organizational and social context. Our major interest here lies, however, in the process of aligning. It is our contention that the distinctive feature of ISD methods and approaches is that they handle or at least recognize this alignment process. We note a number of ISD approaches and traditions with their numerous method instances that explicitly or implicitly focus on the organizational alignment of ISs (Iivari *et al* 1998). We believe that ISD approaches and methods provide an action-oriented framework for condensing and packaging the IS community's collective understanding of organizational alignment and requirements construction.

3.2. A Four-Tiered Framework for ISD Process Knowledge

Elsewhere (Iivari et al 2000), we introduced a four-tiered framework for ISD process knowledge (Figure 2). The framework offers a vehicle for assimilating ISD techniques, methods, approaches, and paradigms, noting that each includes valuable knowledge about ISD. The proposed framework allows for a gradual concretization of ISD process knowledge. Paradigms may be concretized into approaches and approaches into methods with constituent techniques.

The four-tiered framework forms only a skeleton for action-oriented ISD process knowledge. It does not include all possible knowledge constituents of Figure 1. It can, however, be supplemented with a number of constituents of ISD process knowledge. It may include references to facts (e.g. about the applicability of a specific technique to a specific problem), case histories (e.g. successful cases of an application of specific methods and approaches), metaphors (e.g. architecture as a metaphor of ISD). Even though there are published cases of failed ISD projects, they do not appear to have been systematically catalogued. Their linking with the existing knowledge structure of paradigms, approaches, methods and techniques might shed new insights and help us avoid such problems in the future. Techniques, methods, approaches and paradigms may also be backed up with theories. An important point is that each knowledge constituent can be associated with the appropriate unit of the framework. The supplementary knowledge may also be comparatively analyzed with the classes of "ISD paradigms", " ISD approaches", "ISD methods" and "ISD techniques".

---

1 In order to distinguish this from technical implementation, as "implementation" is usually understood in Software Engineering, we use the adjective "organizational".

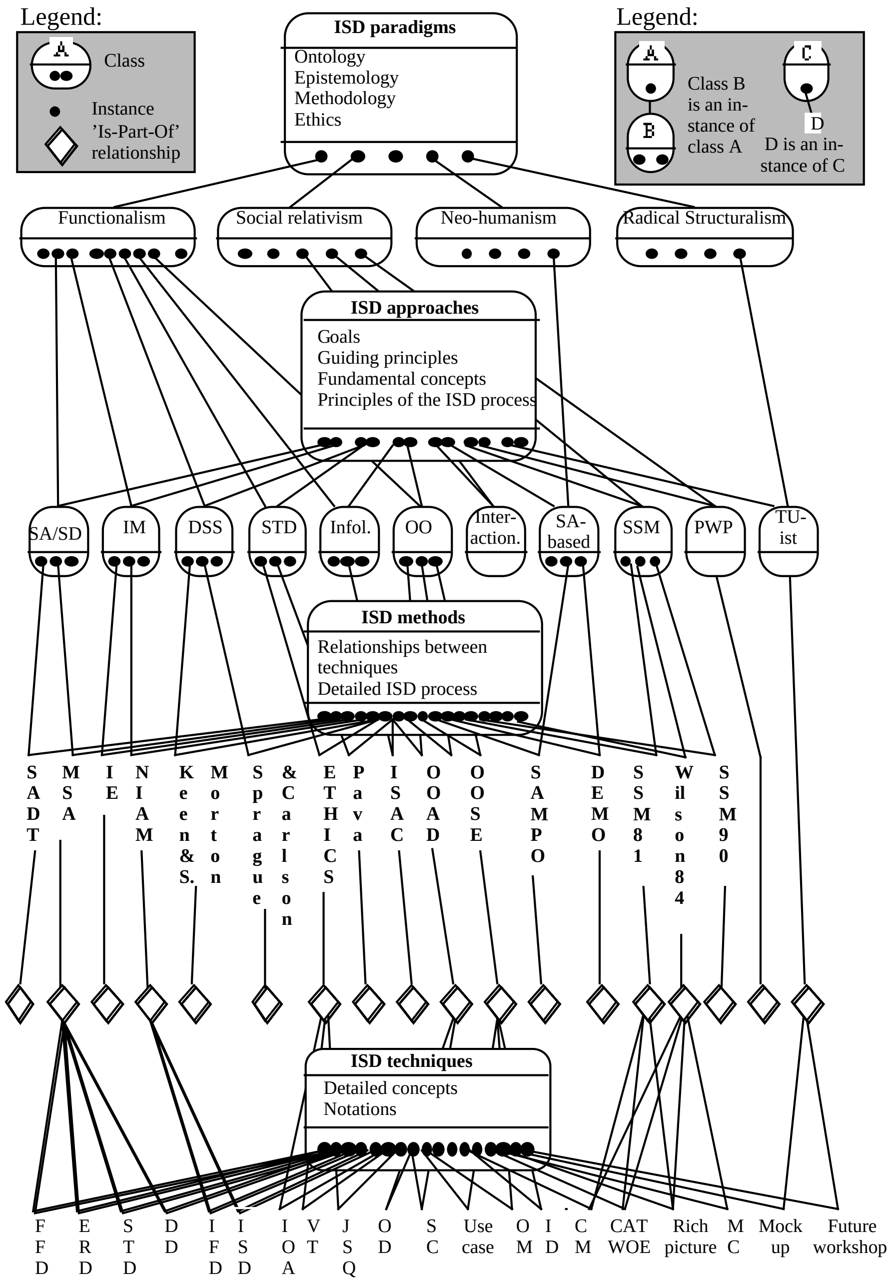

***Figure 2:** Hierarchy of ISD paradigms, approaches, methods and techniques*

4. ISSUES IN ADAPTING ISD PROCESS KNOWLEDGE

4.1. The Nature of ISD as Knowledge Work

In line with Abbott's (1988) characterization of professions, we recognize that knowledge workers do not apply their BoK and skills to the cases at hand in a routine fashion. This is also so in the case of ISD. One of the major research findings into ISD method use, is that methods are not used literally in practice, but ISD methods must be adapted to organization-specific and project-specific contingencies. This suggests the need for method engineering involving the combination of pre-tested method components, fragments or techniques (Kumar and Welke 1992).

The need to customize and adapt ISD methods can be explained by the variety of systems (objects of works) and ISD situations. Perrow's (1967) categorization of organizational technology provides a possible link for understanding the application of the BoK in ISD. He bases his classification on two dimensions: the number of exceptional cases encountered in work and the analyzability of problems, defining the latter as the nature of the search process when an exception occurs. Daft and Lengel (1986) label these two dimensions analyzability (A) and variety (V) of work. When the work process is analyzable, the actors typically follow an objective procedure to resolve the problem. When analyzability is low, there is no exact procedure and work is based more on judgement and experience.

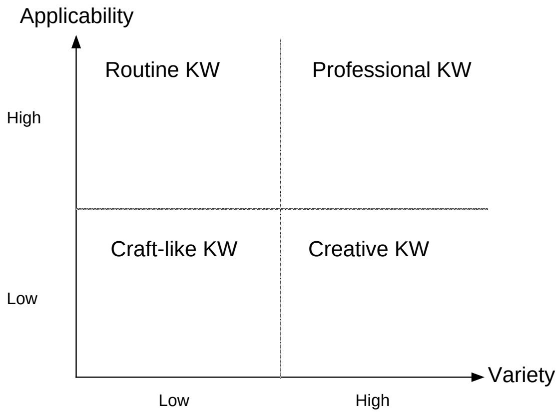

Figure 3: *Classification of KW*

Figure 3 describes the resultant classification, when applicability is substituted for analyzability. Low applicability means that the knowledge must be generalized or made more concrete to suit the situation at hand. The concept of 'variety' (as the variation in the cases and situations to be worked or artifacts to be designed by an actor) allows the BoK to be applied to the concrete objects of work at hand. High variety means that the BoK, if not general enough, must be generalized to cover the novel cases, and then made more "solid" to the required level of concreteness. In contrast to Perrow's (1967) terminology, KW with low applicability and high variety in Figure 3 is labeled *creative KW* while KW with high applicability and high variety, *professional KW*. This renaming is done because engineering work does not only consist of tasks with high applicability of the BoK and high variety of objects of work, but may also include routine tasks, craft-like tasks, and creative tasks. KW with high applicability and low variety is referred to as *routine KW* while KW with low applicability and low variety is termed *craft-like KW*.

As the above discussion suggests, KW is not necessarily homogenous but different tasks may differ in their nature. For example, configuration management in ISD may be thought to be more routine than the visioning of the new work practice and the system to support it. The systems, their parts and situations also may differ. Some systems or their selected parts may be relatively well understood so that there are well-known patterns for how to structure them. Users of the systems also may differ. Some users may have a well-formed understanding of their needs while others may have difficulty to express what they expect from the system. Many of these contingencies are not known from the outset, but they will be encountered during the ISD process. This implies that adaptation of ISD process knowledge must be continuous.

Figure 3 captures two opposite dimensions: craft vs. science (professional) and routine vs. creative which have also been encountered in the context of systems development. Shaw (1990) interprets the development of engineering disciplines as the evolution from craftmanship towards professional engineering, while Ebert (1997) sees that engineering (and ISD) has aspects of both science and craft. Referring to Couger (1996), Cooper (2000) underlines that creativity can be important in all aspects of ISD, while Shaw (1990) emphasizes the significance of routine work as a part of good engineering practice. As hinted above increased knowledge and new technology may change the nature of KW in a specific field. With increased applicability of the BoK, one can expect the general change in KW to be towards more professional and routine work. ISD methods, techniques and tools exemplify this trend.

Figure 3 also implies that the applicability of ISD process knowledge defines the nature of ISD (or its specific tasks) as routine, craft-like, professional and creative KW. Referring to the four-tiered framework (Iivari *et al* 2000) and Table 1, we postulate that *techniques* and *methods* (as an integrated system of techniques) are most relevant in addressing relatively routine aspects of ISD, such as documentation. *Approaches* mainly represent professional level knowledge and *paradigms* mainly support the creative level. If this view of ISD methods and techniques is accepted, it implies that method engineering as a combination of techniques is confined to address relatively routine aspects of ISD.

4.2. Explicit, Tacit and Embedded Knowledge in IS Development

Application of the BoK always requires tacit knowledge (Polanyi 1962). The distinction between the BoK, which is codified, and tacit knowledge is very similar to the “epistemological” distinction of Nonaka and Takeuchi (1995). Their distinction is between *tacit knowledge* that is personal, context-specific, and therefore hard to formalize and communicate, and *explicit knowledge* that is "codified" knowledge transmittable in a formal, systematic language. According to Nonaka and Takeuchi, tacit knowledge includes "mental models" in which human beings create working models of the world by making and manipulating analogies in their minds as well as concrete know-how, crafts, and skills.

In addition to tacit knowledge, organizations can be expected to have embedded knowledge (Badaracco 1991). They possess local knowledge, “complex system of shared information, including abstract models of reality and methods of problem-solving related to technology, which is not formalized but is created spontaneously among work group members and is used by group members to support the performance of work tasks” (Baba 1990, p. 58). Organizations may also have emergent knowledge or knowledge creation capabilities which lie in the social structure and organizational interaction.

	Body of knowledge	*Tacit and embedded knowledge*		*Application processes*
		Individual	*Collective*	
Creative KW	Philosophies Visions Metaphors	Individual creativity	Collective creativity	Intuition Imagination Improvisation
Professional KW	Theories Approaches Strategies	Sensitivity	Absorptive capacity	Adaptation Application Judgement
Craft-like KW	Cases Patterns	Individual skills	Mutual inter-personal skills	Recognition, Comparison Imitation
Routine KW	Techniques Facts	Individual routines	Organizational routines	Routinization Habitualization Automation

Table 1: *Explicit body of knowledge and tacit knowledge in the four categories of KW*

Table 1 attempts to characterize the nature of KW in greater detail, making a distinction between the codified BoK on the one hand and tacit knowledge at the individual level and embedded knowledge at the collective level (see Iivari and Linger (1999) for details). The central constituents of the BoK of *professional KW* are theories, approaches and strategies that always require tacit knowledge, and sensitivity to situational factors. On the other hand, *routine KW* has techniques and facts directly applicable to the extent that the application may be totally automatic in the sense that one does not pay any conscious attention to the task. In *craft KW* there are no clear rules and techniques that can be directly applied making it essentially skill-based, where skills refer to expertise which can be learned only through apprenticeship and practical experience. The BoK may consist of cases and patterns that provide clues for carrying out the task but the challenge for craft KW is the recognition of relevant cases to the problem at hand. *Creative KW* is the least understood and the most difficult to analyze among the four categories of KW. Philosophies, visions and metaphors are useful constituents of the codified BoK in creative KW.

Individual performance especially in programming tasks has been widely studied (Schenk *et al* 1996). In contrast, the performance of systems development teams and organizations has been much less investigated, especially the question of what makes some teams perform considerably better than others.

5. EDUCATIONAL IMPLICATIONS

This paper has hopefully suggested to the reader the value of thinking about what our field does within the context of professionalization. More specifically, our paper has proposed the conception of ISD as KW as an avenue to proceed towards a more "professional" ISD practice. This has obvious educational implications which can be divided into four points.

1. The Body of Knowledge of IS developers should be defined.

This paper proposed six knowledge areas: technical knowledge, application domain knowledge, organizational knowledge, application knowledge, and ISD process knowledge

2. The distinctive competence areas of IS developers should be defined as the core for the education of IS experts.

This paper proposed two areas: IS application knowledge and ISD process knowledge, the latter containing competencies of:

- organizational alignment of IT artifacts,
- user requirements construction,
- organizational implementation of IT artifacts and related organizational changes, and
- evaluation of IT artifacts and related changes.

The paper also outlined a framework for ISD process knowledge that distinguishes ISD paradigms, approaches, methods and techniques. It also underlined that it is the abstract knowledge, i.e. paradigms and approaches that are most distinctive to professional ISD.

3. Education of IS developers should pay proper attention to the variety of tasks in ISD

The paper proposed a framework identifying the nature of ISD as routine work, craft-like work, professional work and creative work.

4. Education of IS developers should pay proper attention to tacit skills and embedded knowledge in ISD

This requires practice-oriented teaching.

REFERENCES

Abbott, A., The System of Professions: An Essay on the Division of Expert Labor, The University of Chicago Press, Chicago, 1988

Baba, M.L., Local knowledge systems in advanced technology organizations, in Gomez-Mejia, L.R. and Lawless, M.W. (eds.), Organizational Issues in High Technology Management, JAI Press Inc, Greenwich, CT, 1990, pp. 57-75

Badaracco, J., The Knowledge Link: How firms compete through strategic alliances, Harvard Business School Press, Boston, 1991

Beyerlein, M.M., Johnson, D.A. and Beyerlein, S.T. (eds.), Knowledge Work in Teams, JAI Press, Greenwich, CT, 1995

Brancheau, J.C., Janz, B.D. and Wetherbe, J.C., Key issues in information systems management: 1994-1995 SIM Delphi results, MIS Quarterly, 20(2), 1996, pp. 225-242

Brooks, F. Jr., No silver-bullet, Essence and accidents in software engineering, IEEE Computer, April, 1987 pp.10-19

Cooper, R.B., Information technology development creativity: A case study of attempted radical change, MIS Quarterly, 24(2), 2000, pp. 245-276

Couger, J.D., Creativity and Innovation in Information Systems Organizations, Boyd & Frazer Publishing Company, New York, 1996

Curtis, B., Krasner, H. amd Iscoe, N., A field study of the software design process for large systems, Communications of the ACM, Vol. 31, No. 11, 1988, pp. 1268-1287

Daft, R.L. and Lengel, R.H., Organizational information requirements, media richness and structural design, Management Science, 32(5), 1986, pp.554-571

DeLone, W.H. and McLean, E.R., Information systems success: The quest for the dependent variable, Information Systems Research, 3(1), 1992, pp. 60-95

Ebert, C., The road to maturity: Navigating between craft and science, IEEE Software, 14(6), 1997, pp.77-82

Feltovich, P.J., Spiro, R.J. and Coulson, R.L., The nature of conceptual understanding in biomedicine: The deep structure of complex ideas and the development of misconceptions, in Evans, D.A. and Patel, V.L. (eds.), Cognitive Science in Medicine: Biomedical Modeling, The MIT Press, Cambridge, MA, 1989, pp.113-172

Ford, G. and Gibbs, N.E., A mature profession of software engineering, Technical Report CMU/SEI-96-TR-004, Software Engineering Institute, Carnegie Mellon University, Pitsburgh, 1996

Freeman, P., Software Perspectives, The Systems is the Message, Addison Wesley, Reading MA, 1987

Freidson, E., Professions of Medicine: A Study of the Sociology of Applied Knowledge, The University of Chicago Press, 1988 (First published 1970)

Freidson, E., Professional Powers: A Study of the Institutionalization of Formal Knowledge, University of Chicago Press, Chicago, 1986

Hardy, C.J., Thompson, J.B. and Edwards, H.M., The use, limitations and customization of structured systems development methods in the United Kingdom, Information and Software Technology, 37(9), 1995 pp.467-477

Iivari, J. and Hirschheim, R. Analyzing information systems development: a comparison and analysis of eight IS development approaches. Information Systems, 21(7), 1996, pp. 551-575

Iivari, J., Hirschheim, R. and Klein, H., A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies, Information Systems Research, 9(2), 1998, pp. 164-193

Iivari, J., Hirschheim, R. and Klein, H., A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches, Journal of Management Information Systems, 17(1), Winter 2000-2001, pp. 179-218

Iivari, J. and Linger, H., Knowledge Work as Collaborative Work: A Situated Activity Theory View, Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999

Jones. M. and Walsham, G., The limits of the knowledge: Organizational and design knowledge in systems development, in Kensall, K.E. (ed.), The Impact of Compuetr Supported Technologies on Information Systems Development, Elsevier Science Publishers B.V (North-Holland), Amsterdam, 1992, pp. 195-213

Kumar, K. and Welke, R.J. Methodology engineering: A proposal for situation specific methodology construction. In Cotterman, W.W. and Senn, J.A. (eds.). Challenges and Strategies for Research in Systems Development. John Wiley & Sons, Chichester, 1992, pp. 257-269

Macdonald, K.M., The Sociology of the Professions, Sage Publications, London, 1995

Machlup, F., The Production and Distribution of Knowledge in the United States, Princeton University Press, Princeton, NJ, 1962

McConnell,S. and Tripp, L., Professional software engineering: Fact or fiction, IEEE Software, 16(6), 1999, pp. 13-17

Minzberg, H., Structures in Fives, Designing Effective Organizations, Prentice-Hall,.Englewood Cliffs, NJ, 1983

Nonaka, I. and Takeuchi, H., The Knowledge Creating Company, Oxford University Press, New York, 1995

Orlikowski, W. and Baroudi, J., The information systems profession: Myth or reality, Office: Technology and People, 4(1), 1989, pp. 13-30

Perrow, C., A framework for the comparative analysis of organizations, American Sociological Review, 32(2), 1967, pp.194-208

Polanyi, M., Personal Knowledge: Towards a Post-Critical Philosophy, Chicago University Press, Chicago, IL, 1962

Reich, B.H. and Benbasat, I., Factors that influence the social dimension of alignment between business and information technology objectives, MIS Quarterly, 24(1), 2000, pp. 81-113

Schenk, K.D., Vitalari, N.P. and Davis, K.S., Differences between novice and expert systems analysts: What do we know and what do we do?, Journal of Management Information Systems, 15(1), 1998, pp.

Schultze, U., A confessional account of an ethnography about knowledge work, MIS Quarterly, 24(1), 2000, pp. 1-41

Shaw, M, Prospects for an engineering discipline of software, IEEE Software, 7(6), 1990, pp.15-24

Smithson, S. and Hirschheim, R., Analysing information systems evaluation: Another look at an old problem, European Journal of Information Systems, 7(3), 1998, pp. 158-174

Swanson, E. B., Information System Implementation, Bridging the Gap between Design and Utilization, Irwin, Homewood, Il, 1988

SWEBOK, Guide to the Sofware Engineering Body of Knowledge, A Stone Man Version (Version 0.6), February 2000 (www.swebok.org)

SWECOE, Software Engineering Code of Ethics and Professional Practice, (www.cs.etsu.edu/seeri/secode.htm)

Vitalari, N.P., Knowledge as a basis for expertise in systems analysis: An empirical study, MIS Quarterly, 9(3), 1985, pp. 221-241.

Wynekoop, J.L. and Russo, N., Systems development methodologies: Unanswered questions and the research-practice gap, in DeGross, J.I., Bostrom, R. and Robey, D. (eds.), Proceedings of the Fourteenth International Conference on Information Systems, Orlando, FL, 1993, pp.181-190

Zwerman, W.L., Profession/occupation without a history, IEEE Annals of the History of Computing, 21(1), 1999, pp. 66-70

Zuboff, S., In the Age of the Smart Machine, The Future of Work and Power, Heineman, Oxford, 1988