FULL REFERENCE:

- Klein, J. & **Eseryel, D.** (2005). The corporate learning environment. In M. D. Lytras & Ambjörn Naeve (Eds.), *Intelligent learning infrastructures for knowledge intensive organizations: A semantic web perspective* (pp. 1- 38). Hershey, PA: IDEA Group Publishing.

Chapter I

The Corporate Learning Environment

Jerry Klein, KnowledgeKraft, Inc., USA

Deniz Eseryel, Syracuse University, USA

Editors' Notes

Jerry and Deniz give an excellent overview of the corporate learning environment. We found their deep understanding and thorough analysis very informative. Their clear and lucid presentation of the field concludes with a framework for corporate learning environments. Within its context, a number of critical issues are discussed and readers are asked to use their critical thinking to further exploit the communicated meanings.

We invite you to comment on their proposition about the required actions and steps for developing the corporate learning environment. At the end of this chapter, a critical shift in your thinking will be evident. You will require more guidance on how socio-technical systems and applied informatics can support such a framework in real-world situations.

Since you will acknowledge that Jerry and Deniz describe the rich picture for the requirements and the directions toward corporate learning environments, you will be thirsty for the contents of the next chapter, which incrementally contributes to our vision for the learning organization of the 21st century.

Many additional resources and a case study are provided at the end of the chapter. We are certain that you could spend many hours with these materials, and we encourage you to do so. The authors will be delighted to receive your comments.

Abstract

Emerging technology has changed the focus of corporate learning systems from task-based, procedural training to knowledge-intensive problem-solving with deep conceptual learning. In addition, the deployment of open systems and distributing processing are adding new stresses to learning systems that can barely keep pace with the current rate of change. Learning environments to address these challenges are viewed within a framework of the conventional learning curve, in which different learning elements are required to support different levels of expertise. An adaptive development model for creating and sustaining a learning environment is proposed that consists of the iterative application of three phases: (1) analysis and reflection, (2) architecture inception and revision, and (3) alignment. The model relies on the notion that analysis deals as much with synthesis and learning as it does with decomposition. We conclude that the concept of a "learning environment" provides a viable construct for making sense of the array of systems designed to support knowledge management, document management, e-learning, and performance support. A learning environment with a well-defined architecture can guide the convergence of multiple systems into a seamless environment providing access to content, multimedia learning modules, collaborative workspaces, and other forms of learning support. Finally, we see future learning environments consisting of networks of databases housing content objects, elegant access to the content, ubiquitous virtual spaces, and authoring tools that enable content vendors, guilds, and universities to rapidly develop and deliver a wide range of learning artifacts.

Introduction

In recent years, the importance of knowledge as a source of sustainable competitive advantage has been discussed by a myriad of authors (Drucker, 1993; Leonard-Barton, 1992; Nelson, 1991; Nonaka & Takeuchi, 1995; Prahalad & Hamel, 1990). In a knowledge-based economy, the new coin of the realm is continuous learning. Today's companies have been exhorted to consider knowledge creation a source of competitive advantage by building a corporate learning environment that focuses on the requirements of knowledge workers in order to meet the demands of the post-industrial information economy. As Reich (1998) notes:

Want to build a business that can outlive its first good idea? Create a culture that values learning. Want to build a career that allows you to grow into new responsibilities? Maintain your hunger to learn — and join an organization where you will be given the chance to learn continuously. (p.198)

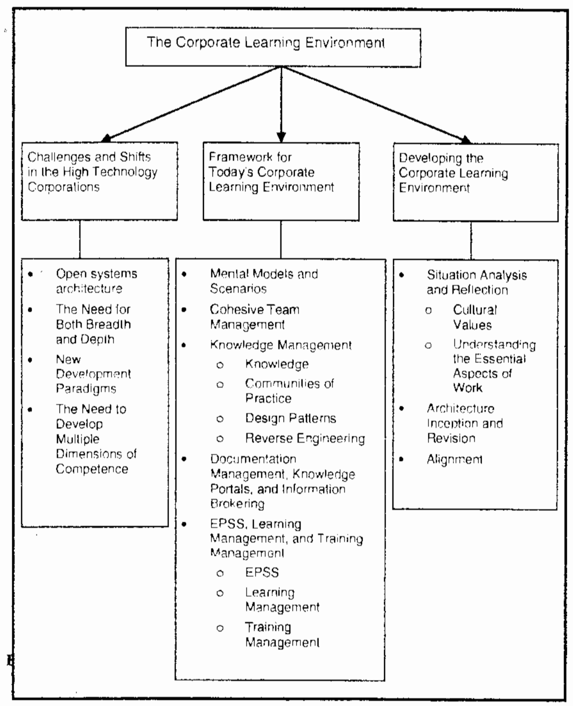
This chapter presents a conceptual framework for creating and sustaining a viable corporate learning environment (see Figure 1). This framework integrates the various

human learning systems operating within corporations engaged in product development. These systems include cohesive team management, knowledge management, documentation management, performance support, learning management, and training management. This chapter first provides an overview of the changing nature of the corporations and identifies the challenges that these changes bring. Then, the framework for today's corporate learning environment is discussed in detail. Next, a high-level model for developing and continuously improving a corporate learning environment is presented. This chapter concludes with a discussion on future directions.

Challenges and Shifts in the High Technology Corporations

Technology and globalisation have resulted in small market windows, rapid changes in products, and geographically-dispersed work groups. These changes have caused basic shifts in core processes and organisational structures which have placed new demands on the corporate learning environment. Some of the more significant changes and their implications are discussed here.

Figure 1. Chapter overview



Open Systems Architecture

Open systems and distributed proeessing have created havoc in corporate product development processes and created new demands on the learning environment. Open systems architecture enables parallel development and products to evolve at a rapid pace. In addition, distributed processing results in systems of such complexity that they are virtually impossible to comprehend at any level of depth by a single individual. Systems are designed to evolve, and engineering, to a large extent, involves modifying and enhancing the existing system. This in turn requires engineers to understand how complex systems work. Consequently, learning has become the core activity of most design engineers engaged in systems development. For instance, two studies at Bell Laboratories found that software developers devote over 50% of their time in discovery and learning (Klein, 1995).

Complex open systems have resulted in expertise shifting from the front end to the back end of the product development process. While much of the routine first-tier maintenance work has been reduced because these systems can diagnose their own faults, when failures and problems do arise they require a person with deep understanding to diagnose and resolve. Consequently, staff in upper-level tiers of the maintenance hierarchy have developed detailed, deep, and broad system knowledge. On the other hand, staff on the front end, such as system engineers and architects, work at a high conceptual level with only a cursory understanding of implementation details. This has resulted in training and documentation developers now relying on maintenance staff as the key source of information rather than system architects. It has also resulted in a major shift in the focus of training and documentation from that of supporting procedural learning to supporting knowledge-intensive problem-solving.

Technical training and documentation have traditionally been focused on routine task-based maintenance training, but conditions now require training to focus on developing deep knowledge. This situation is a result of the automation of first-tier maintenance tasks and complex problem-solving at the higher maintenance tiers, and also is due to the tremendous increase in the size of development engineering staff. Open systems enable organisations to have hundreds of designers working in parallel, and development staff on some projects have grown into the thousands. Corporations face the problem of providing learning support for them. This shift in learning goals requires high technology corporations to define and develop complex skills, in addition to procedural skills, understanding and defining the mental processes involved in the comprehension of complex systems rather than simply specifying and decomposing routine tasks, and requires learning environments to help both design engineers and maintenance engineers assimilate massive amounts of complex information (Eseryel & Spector, 2000).

The Need for Both Breadth and Depth

All phases of product development — from initial conceptualisation to manufacturing, and to deployment and support — require depth and breadth of knowledge. The primary method for meeting this requirement is through the use of teams that span domains,

organisational functions, and geographical distance. With respect to product design, corporations such as IDEO rely on teams of people with diverse backgrounds to generate innovative and successful products rather than relying on the "lone genius in the lab." IDEO has created a work environment and design processes that exploit the talents, experience, and insights of people from multiple fields that range from engineering to anthropology (Kelley, 2001).

In addition to using teams in the design phase, corporations use concurrent engineering methods involving teams of staff from research, engineering, manufacturing, and maintenance. The underlying principle is that by collaboratively developing manufacturing processes concurrently with the design of the product, corporations can optimise the product lifecycle and design a product that can be produced as efficiently as possible (Miller, 2003).

Bringing this wide range of expertise to the product lifecycle is a key challenge in today's corporate world. The collaborative and concurrent engineering practices have resulted in a need for learning environments to support the "collaborative creation, management, dissemination, and use of product definition information across the extended enterprise from early conceptual stages through design, production, maintenance, and field support" (Miller, 2003, paragraph 7). The situation also requires leaders with perspective and breadth of knowledge who can deal with ambiguity and who are able to establish processes to support a sharing community. Developing this leadership is a critical challenge. A common strategy for creating effective multi-disciplinary teams is co-locating people in close proximity with one another. However, today's corporations have product teams geographically distributed across the globe. Therefore, an essential challenge to the learning environment is to create a "virtual space" for teams that provides the same supportive attributes of close physical space.

Supporting teams is crucial in order for corporations to thrive. However, most aspects of the traditional learning environment are focused on the individual. Recognition and reward policies, organisational structures, work processes, training, and documentation are normally directed toward rewarding, developing, and supporting the individual. It is critical that the learning environment be redesigned to support team learning and development.

New Development Paradigms

New technologies and rapid changes in the global market have resulted in corporations using new processes and, in some cases, new paradigms in creating products. These strategies are driven by a need to reduce time-to-market and a need to customise products. Three prevalent design paradigms used in today's corporations include: (1) automating the traditional top-down systems approach, (2) incremental development, and (3) customer creation environments.

The systems approach to development is well-documented and, we assume, familiar to the reader. The central idea of the systems approach is to make the basic processes work faster and better by automating and improving each step in the process. Incremental development, on the other, hand represents a marked departure from the traditional systems approach and goes under a variety of names such as agile development, adaptive development, rapid prototyping, or spiral development. While the approach is primarily currently used for developing software, it can also be applied to a variety of contexts. The key distinctions between the two approaches can be summarised as follows (Ambler, 2003; Larman, 2001):

1. *Design and development is incremental and iterative.* Traditional development uses a waterfall or sequential lifecycle in which all requirements are first defined, then followed by development, integration, and testing. In the iterative approach, development is organised into small mini-projects of short time periods, each of which produces a tested, integrated, and functioning system. Overall product development consists of successive enlargement and refinement of the system through multiple iterations.
2. *Feedback drives the process rather than full and complete specifications.* In traditional development methods, requirements are specified and frozen — then development begins. Any change in requirements creates havoc in subsequent development phases. In the iterative paradigm, change is expected and embraced. Consequently, the product's architecture and the development environment are designed to support change, rapid development, and testing.
3. *Close customer involvement during development.* The traditional approach normally organises people by function. One organisation does the front-end phase of meeting with customers and developing requirements, which are then passed along to the development organisation. Iterative development staff, on the other hand, works directly with customers as they design and develop the product as they have the main responsibility for both deriving requirements and developing the solution. The rationale is that: (1) written specifications do not communicate very well, (2) customers are not sure what it is they want and are poor at articulating requirements, (3) translations and intermediaries between the customer and those designing the product often result in confusion, and (4) one can leverage the expertise of the development staff, since they are the people most familiar with the technology.
4. *Iterative development focuses on high-risk, high-value features first.* The most difficult features are tackled first in the incremental development approach. The basic idea is that if one cannot solve the most complex and critical problems, then one should not continue. Larman (2001) illustrates this point with the following anecdote: "If I want Web pages to be green and the system to handle 5,000 simultaneous transactions, green can wait" (p. 37).

Customer development environments represent a third paradigm for creating products. Under this approach, customers are given the means to design and create their own product. Like a salad bar, which lets the customer construct their own salad to suit their tastes, development tools are provided that enable the customer to create their own product. An early example of this is the Advanced Intelligent Network and Customer Service Creation Environment developed and deployed by Bell Laboratories in the 1980s.

This system enables regional phone companies to develop and offer new custom solutions rather than having Bell Lab's development staff build the features. Recent research by Thomke and von Hippel (2002) has shown the benefits of providing "innovation toolkits" that enable customers to design customised products for themselves.

Implications of these development strategies to the learning environment are profound. All the paradigms result in rapidly changing products and variable product designs. This factor alone causes a great deal of difficulty in organisations using processes designed to produce documentation and training on a stable and "standard" product. Vendor, customer field support, and maintenance staff experience a great deal of confusion when faced with multiple versions of the product, and internal development staff themselves have difficulty keeping up-to-date on the latest version of the product. Consequently, a critical problem is keeping documentation and training in synch with product development.

The Need to Develop Multiple Dimensions of Competence

Traditional learning environments have focused on technical competence and domain-specific knowledge. Other skills, such as effective communication, teamwork, and tolerating diversity, are often referred to as "soft skills" and usually have low priority. The new development paradigms and work processes have moved these "soft skills" up the list of priorities. The roles of the staff have changed. Consequently, the learning environment must reflect these changes. For example, the animosity between marketing/sales and engineering, which seems to be a tradition in most organisations, cannot be tolerated. Sales staff motivated by "closing the deal" and engineers motivated by creating technical solutions have to reconcile their perspectives and collaborate. Staff must work together across functional entities within the organisation and learn to work with a variety of people as they deal with customers, colleagues, partners, and vendors spread across the globe. People are now required to have the skills, talents, and traits that will enable them to cohesively work together and deal with ambiguity in performing the most basic tasks. We identify four areas of competence the learning environment should support: domain-specific knowledge, instrumental skills, cognitive strategies, and traits. Domain-specific knowledge includes the traditional areas of content expertise and skills. Instrumental skills refer to the skills needed in order to work effectively in social situations, which involve knowing how to get things done in the company. Cognitive strategies are the mental processes we use for reasoning, thinking, and solving problems. Traits are attributes of the personality such as motives, interests, creativity, and initiative.

All the dimensions of competence can be defined, developed, and supported. For instance, the product development division at Bell Labs' Network Systems identified the instrumental skills and traits that are necessary to be a "star performer." These skills included organisational savvy, networking, perspective, self-management, and teamwork effectiveness. What is interesting is that just defining and publishing a checklist of the skills made a significant difference in staff performance (Kelly & Caplan, 1993).

Summary and Implications

Substantial efforts and commitment must be made to establish and maintain an effective corporate learning environment. Emerging technology has changed the focus of learning systems from task-based, procedural training to knowledge-intensive problem-solving that involves deep conceptual learning. The deployment of open systems and distributing processing are adding new stresses to learning systems that can barely keep pace with the current rate of change. Work today not only requires technical knowledge but the instrumental skills that enable people to work cohesively across disciplines and global borders. These emerging requirements and challenges are addressed in the framework we propose for creating a corporate learning environment, which is presented next.

Framework for Today's Corporate Learning Environment

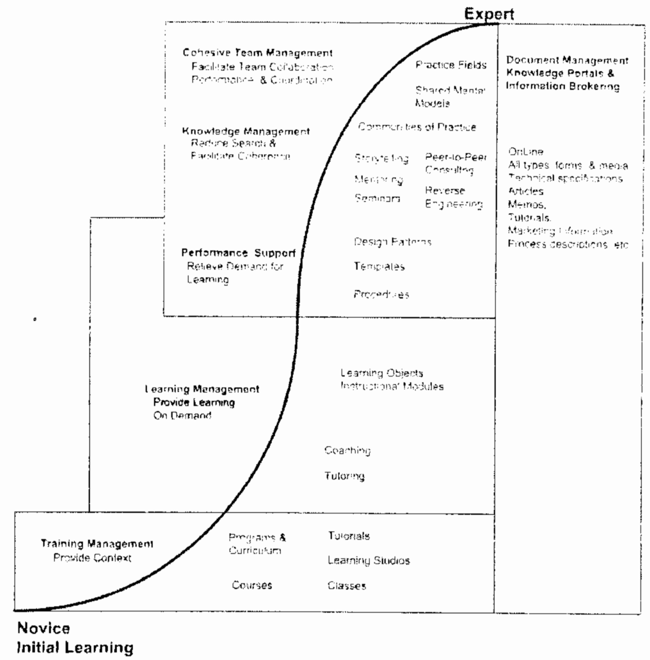
The corporate learning environment is an integrated mosaic of systems, tools, and processes facilitating individual and team learning, performance, and development. This mixture of systems can be viewed within a framework based on the premise that different methods are needed for different levels of knowledge and expertise. The framework (see Figure 2) is an extension of a model presented by Dillon and Hallett (2001). It applies the notion of the conventional learning curve to the context of the corporate learning environment in order to supply an apt structure for understanding when and how different modes are employed.

Within this framework, the conceptualisation of the learning environment consists of systems to manage and support: (1) cohesive team management, (2) knowledge generation and sharing, (3) performance support, (4) document storage and retrieval, (5) on-demand learning, and (6) traditional training. In addition to discussing each of these systems, the concept of mental models is discussed because the corporate learning environment must be based on what is known about how people learn. The notion of mental models is a construct from the field of human learning that can be used as a core concept in guiding the design of learning environments.

Mental Models and Scenarios

The construct of a mental model and the corresponding scenarios used to develop them can be a unifying concept in designing the overall learning environment. Effective teamwork depends on a shared mental model (Edmondson, 2003; Senge, 1990). Research has shown that people mentally construct and "run" mental models in performing a wide range of tasks including reading (Kintsch, 1986), medical diagnoses (Kuipers & Kassirer, 1984), software design (Soloway, 1986), and troubleshooting complex systems

Figure 2. Framework for corporate learning environments



(Rasmussen, 1986). Rouse and Morris (1985) assert that "... it is difficult to explain most aspects of human behaviour without resorting to the construct of mental models" (p. 1). The cognitive processes involved in using mental models to comprehend text are demonstrated with the following example adapted from Galamos (1986). In order to get the true effect for the process, read the line in Box 1 and then read the next lines in Boxes 2 and 3 later in the chapter.

Box 1

The pirates blindfolded the victim and started him on his walk.

Go to Box 2.

In reading the three sentences in Boxes 1, 2, and 3, the first line most likely evoked an image in your mind of a ship with pirates walking a person down a plank and you would expect the next line to refer to those elements. Some confusion may have set in when you read the second line. However, after reading the last line, the second sentence "made sense." According to Galamos (1986) and Kintsch (1986), we construct and modify a situation model of the text as we read. In this example, the first line would elicit a model that fits the text. We then use top-down processes to assimilate the new text into the model. When we come across the second line, top-down processing falters and we become confused. When we read the third line, bottom-up processing takes over, which involves constructing a new model to fit the text; the pirate cruelty model is replaced with the masquerade party model.

The various elements of the learning environment should be focused on helping people construct coherent models. The knowledge worker is swamped with vast amounts of information and is in a constant mode of learning. The learning environment must help them mentally stitch together knowledge from multiple sources. We call this process "knowledge assembly" in which people make sense of what is happening by assembling information into a coherent whole. Training must establish the base mental models on which a person can then learn subsequent knowledge through elaboration. Knowledge management techniques, such as reverse engineering and low fidelity simulations, can be designed to facilitate the development of mental models of complex devices. Documentation must provide structured information that enables the person to develop deeper models as they need to learn more details, and team development and coordination methods should be designed to support the building of shared mental models.

Scenarios are a powerful method for facilitating the development of mental models and deep conceptual learning. The term "scenarios" has a variety of connotations. In our context, we use the term to denote the mental execution of real or imagined sets of events in either hypothetical or real situations (e.g., illustrating how the Web works by walking through the sequence of events that occur in each node in a network or constructing an explanation of why a bridge collapsed by identifying the sequence of events that caused the failure).

To the penthouse

Box 2

Go to Box 3.

In dealing with complex systems, people walk through scenarios in order to clarify and confirm their understanding (Bennet & Morgan, 1987; Klein, 1995; Soloway, 1986). In product development, engineers formulate and revise scenarios in order to define functionality and to clarify system specifications with end users and stakeholders (Weidenhaupt, Pohl, Jarke, & Haumer, 1998). Scenarios are also an element of standard modeling languages (Cockburn, 1997).

Scenarios are effective communication tools and are spontaneously created in the day-to-day activities in most business practices. The formal systems in the learning environment need to exploit these scenarios and build their content around them. However, there is a tendency on the part of knowledge management practices to take scenarios, extract

generalisations and rules, and then create a document or training session of best practices or business rules. This practice adds little learning value to anyone other than the person doing the extracting. You cannot learn something for someone else; each person needs to abstract his or her own principles, rules, and generalisations. As Barron (2000) states:

It is clear you have to capture knowledge in the ways it is being communicated, by the people who are doing the communicating. ... Rather than deconstruct what a top salesperson does to try and create a best practices model that catalogues techniques — an approach that is time-consuming and often overlooks many key intangibles — the new approach would be to capture that person in well-crafted role-playing scenarios ... Doing so would provide learners a more direct insight into the salesperson's skill and allow content to be generated more quickly. (p. 2)

Cohesive Team Management

Effective teamwork is a goal of every corporate manager as working in a cohesive team is an aspiration of most staff members. Senge (1990) eloquently describes the high value people place on teamwork in his classic book on organisational learning where he writes:

Box 3

Where the masquerade party was in full sway.

Most of us at one time or another have been part of a great "team," a group of people who functioned together in an extraordinary way — who trusted one another, who complemented each others' strengths and compensated for each others' limitations, who had common goals that were larger than individual goals, and who produced extraordinary results. I have met many people who have experienced this sort of profound teamwork — in sports, or in the performing arts, or in business. Many say that they have spent much of their life looking for that experience again. (p. 4)

Cohesive team development and management is placed at the apex of our framework in Figure 2 to indicate the significance corporations and individuals place on teamwork. More importantly, team development is placed at the top of the learning curve to illustrate the fact that individual expertise is a critical precondition of high performing teams. Research (Edmondson, 2003; Klein, 1998) and cases studies at corporations such as IDEO (Hargadon & Sutton, 1997) clearly indicate that individual expertise is a core ingredient of effective teams regardless of the context. IDEO, for instance, encourages designers to become experts in a particular product area and learn everything that is out there in that product area, but then bring their expertise to the team of product designers.

Studies by Klein (1998) indicate that firefighters function effectively as a team because each member is highly competent and experienced. Edmondson (2003) found that expertise is a critical characteristic of effective surgical teams.

Other characteristics of effective teams include: (1) having a strong team identity and a shared knowledge that enables members to plan, anticipate, and execute; (2) having members that compensate by stepping outside of their assigned roles in order to help the team; and (3) achieving situation awareness by diverging and converging, which is the process of actively seeking a variety of views from team members, and then converging the views into a coherent whole (Klinger, 2003).

Developing and supporting teams requires a variety of conditions, the first one being psychological safety. The organisation must provide an environment conducive to taking interpersonal risks. Edmondson (2002) notes that change involves interpersonal risk to a person's image because change involves: (1) asking questions and seeking information which creates the risk of being seen as incompetent or ignorant (no one else is asking it, maybe I am supposed to know it); (2) admitting mistakes and asking for help which could result in being perceived as incompetent; and (3) reflecting and getting feedback which could result in being perceived as being negative (providing criticism might be perceived as disruptive). Psychological safety enables people to engage in the interpersonal risky behaviours required for learning (Edmondson, 2002).

Compelling shared goals are another condition for supporting effective teams. During her studies on technology adoption in surgery, Edmondson (2003) found that surgical teams with higher order goals that centred on benefits to the patient (e.g., "This will be less intrusive to the patient.") were more successful than teams in which the goals reflected extrinsic factors (e.g., "You will attain high visibility and be recognised as a leader for your innovative practices."). Teams must have passionate people who are dedicated to achieving higher order goals — something to engage each individual.

One way to establish shared meaningful goals is through self-selecting, self-organizing, and self-managed teams. Software development teams using agile methods are characterised by self-organisation and intense collaboration, within and across organisational boundaries (Cockburn & Highsmith, 2001). Design companies such as IDEO allow staff to place themselves on projects. The power of this facet of teamwork has been stressed by Schrage who claims that innovation is not accomplished by forming good teams (Manasco, 1999). Instead, interesting concepts are proposed or fascinating projects are initiated by individuals forming themselves into a team. Schrage further states (Manasco, 1999):

> *... more often than not, is that innovative prototypes generate innovative teams. It is not that innovative teams generate innovative prototypes. What really happens is that innovative people build a model of something and then they show it to others they think might have an interesting comment. A team forms. The prototype generates a community of interest.* (p. 1)

A key challenge that management face in promoting teamwork is to devise appropriate recognition and compensation structures. We can turn to IDEO to gain insight into this

issue. IDEO's reward system provides substantial support for collaboration. Managers determine designer's pay, and while they place weight on the number of hours billed, compensation decisions are based largely on informal reputation among formal peer reviews. The only way to enhance one's status in the organisation is by earning the respect of his or her peers. Designers earn this respect through individual efforts that produce good designs, but a designer's reputation is based at least as much on using his or her skill to help others. Also, a designer's reputation is enhanced by asking for help. People who do not ask for help are thought to be either too insecure or too arrogant to lack humility about what they know and respect for what others at IDEO know. There is especially low tolerance at IDEO for engineers who do not ask for help and then produce poor designs. Hargadon and Sutton (1997, p. 735) conclude, "[T]he most respected people at IDEO are part pack rat [because they have great private collections of stuff], part librarian [because they know who knows what], and part Good Samaritan [because they go out of their way to share what they know and to help others]."

The essential functions of the learning environment are to create cohesive teams by developing individual expertise, establishing higher-order goals, providing psychological safety, allowing self-management, and aligning reward structures. However, implementing these functions will be for naught unless appropriate space is provided. Team members need to be in contact with each other and space is critical to teamwork. IDEO organises space around the concept of neighbourhoods in order to facilitate spontaneous interaction, and many of the companies practicing agile methods of development deliberately arrange space to foster teamwork. For instance, eXtreme programming methods call for two people to be assigned to the same workstation while working together on a design (Kerievsky, 2001).

In today's global economy, team members are often geographically dispersed around the world, and emerging technology is being used to support virtual collaboration. However, these new tools stress communication over space. Schrage emphasises the need to make the distinction between communication and shared space (Manasco, 1999):

> *The key element, the key ingredient, the key medium for successful and effective collaboration is the creation and maintenance of a shared space. You cannot create shared understandings without shared space. It is not an exchange of memos. It is not a meeting. It is not e-mail. It is not videoconferencing. A shared space is the place or the medium where people put up and play with — in a shared context — the representations and models of their ideas. It can be a blackboard. It can be a whiteboard. It can be Lotus Notes. But it is a shared space — a shared place — where people can manipulate and iterate ideas and representations of the ideas.* (p. 2)

Knowledge Management

Understanding how corporations create new products, new methods, and new organisational forms is important. A more fundamental need is to understand how organisations create new knowledge that makes such creations possible. The corporation that wishes to lead in today's chaotic business environment needs to be the one that creates and manages knowledge effectively. This section discusses the general issue of "what is knowledge" and how knowledge is shared between communities of practice. It then describes the use of design patterns and reverse engineering which are two of the more effective knowledge management methods used in technology intensive industries.

Knowledge

Knowledge is one of those concepts that is extremely meaningful, positive, promising, and hard to pin down. According to cognitive scientists, knowledge involves cognitive structures that represent a given reality. A corporate manager is much more likely to associate knowledge with specific situations and "know-how." This is to show that knowledge is often in the eye of the beholder. People tend to give meaning to the concepts through the way they choose to use them (Wittgenstein, 1958). In the field of knowledge management, though, researchers are mainly focused on two types of knowledge: explicit and tacit (Edmondson, Winslow, Bohmer, & Pisano, 2003; Polanyi, 1966). Explicit knowledge can be put on a paper, formulated in sentences, or captured in drawings. An engineer, for example, conveys his or her knowledge of a product design through drawings and specifications, making what he or she knows explicit. Yet, tacit knowledge is tied to the senses, individual perception, physical experiences, rules of thumb, and intuition. Tacit knowledge is difficult to describe to others. Selecting the best possible move in a very complicated chess position, for example, or interpreting a complex readout of a nuclear reactor demands knowledge that cannot be found in a manual or easily conveyed to a novice.

Polanyi (1966) contends that human beings acquire knowledge by actively creating and organizing their own experiences. Thus, knowledge that can be expressed in words and numbers represents only the tip of the iceberg of the entire body of knowledge. As Polanyi (1966, p. 4) puts it, "We can know more than we can tell." While the idea of tacit knowledge makes intuitive sense, it is often difficult to come to grips with it on a practical level. Recognizing the value of tacit knowledge and figuring out how to use it is the key challenge in any knowledge management initiative.

Knowledge is created only by individuals. An organisation cannot create knowledge. The organisation can only support creative individuals or provide contexts for them to create knowledge. Therefore, effective knowledge creation depends on an enabling context (von Krogh, Ichijo, & Nonaka, 2000). Knowledge is dynamic, relational, and based on human action within a context. Consequently, supporting the whole process of knowledge creation requires the necessary context. Organisational knowledge creation, therefore, should be understood as a "process that 'organizationally' amplifies the knowledge created by individuals and crystallizes it as a part of knowledge network of the organization" (Nonaka & Takeuchi, 1995, p. 59). This process takes place within an expanding "community of interaction" which crosses intra- and inter-organisational levels and boundaries.

Communities of Practice

Sharing knowledge is common within groups where collaboration and teamwork are the norm. Engineers, for example, naturally create and share knowledge as they work together designing or developing new products, during which knowledge movement takes place without any formal mechanism for knowledge capture or retrieval. However, once the view of the organisation expands beyond a functional group we see distinct communities of designers, engineers, operators, and so on. These communities of practice are groups of people whose interdependent practice binds them into a collective of shared knowledge and common identity. Within such tight-knit groups, "ideas move with little explicit attention to transfer, and practice is coordinated without much formal direction" (Brown & Duguid, 2000, p. 88). It is the flow of knowledge between communities that formal processes are required, and the key challenge for knowledge management systems is to facilitate this movement of knowledge between diverse groups while not inhibiting the spontaneous generation and flow of knowledge within the groups.

The quality improvement, process re-engineering, and knowledge management trends that took place during the 1980s and 1990s focused squarely on the movement of knowledge between communities, with documentation was the primary mechanism utilised. The emphasis on process and documentation often had a detrimental effect on the creation and movement of knowledge. Good designers in good companies thrive on networking. They intuitively know that documentation is not the most effective way to communicate knowledge. Designers learn by interacting with others when they apply their knowledge in new contexts and by consulting with each other on projects. But, the mantra of knowledge management was "capture what you know," with the implied benefit of reducing the amount of time devoted to consulting and teaching others. Essentially, knowledge management processes were telling people to document what they know so they would not have to meet, consult, and share knowledge with each other. Management processes were often out of alignment with the natural processes of knowledge creation and flow. This inconsistency can result in disengagement, which, in turn, impedes the generation of new knowledge.

Much of the appeal of the agile development methods is a return to focusing on people rather than on process and documentation. Common knowledge management methods within the agile paradigms include (Jeffries, 2001; Kerievsky, 2001):

1. Making knowledge visible by using white boards, large charts, and entire walls to draw, model, discuss, explain, and display designs.
2. Short tutorials in which designers describe their design to one or two of their colleagues.
3. Repositories of index cards that enable people to quickly jot down their insights as they work.
4. After action reviews in which all members discuss what worked and what to do differently next time.

These methods are designed to facilitate the generation and movement of knowledge within groups. However, with the use of synchronous learning tools, these methods can be used to move knowledge across time, space and the boundaries separating communities of practice. Virtual classroom and meeting tools have the capability to record the audio and visual interactions that take place as people communicate. The white board drawings and short tutorials can be easily recorded and stored for later use. Barron (2000) emphasises the value of this technology:

The primary e-learning tool of the e-learning and KM era will be synchronous e-learning tools that will be used to quickly record and disseminate SME knowledge... Virtual classroom tools will become the knowledge capture tools of the future... (p. 4)

Additional methods to move knowledge across boundaries include using product managers to oversee the entire development cycle and the use of instructional designers dedicated to developing courses that stitch together a coherent picture from multiple information sources.

Design Patterns

Experts have the ability to solve tasks that are not routine and to deal with the unexpected (Dreyfus & Dreyfus, 1986). The difference between experts and novices lies in the ability to select the right materials and discard materials that might be flawed. For instance, an expert chess player recognises the patterns on a chess board and recalls only the very best strategies that work in those cases, and then selects the one that works best (de Groot, 1965). Unfortunately, such expert knowledge is hard to be made explicit and the means for sharing tacit knowledge (observation, narration, etc.) will be difficult to apply. An explicit effort is required to share expert knowledge with novices. In the domain of software engineering, for instance, software patterns are developed to provide a mechanism for rendering design advice in a reference format. Software design is a massive endeavour, and when faced with a design problem, one must be able to focus on something as close to the problem as possible. It is frustrating to find an explanation of what one needs to do, while buried in a large example that contains 20 things that are unrelated but must be understood in order to complete the things relevant to the task. So, software patterns help by trying to identify common solutions to recurring design problems. When experts write software patterns, they not only include how to implement the pattern but also when to implement the pattern as well as the alternative patterns so that novices can make the right selection. Once the right selection is made, then the novice designer finishes the software design process by adapting the pattern to the given problem context. This provides the opportunity for the (rather) novice designer to climb up the ladder of expertise and continuously learn on the job. This also provides the expert designer to share his or her expertise, to gain respect and recognition of his or her colleagues, thereby, to increase his or her job satisfaction.

Reverse Engineering

We can learn and understand a device by having someone explain it to us or by studying the documents. Another method is to take it apart and see how it works — often called reverse engineering. Reverse engineering is prevalent in product design and development organisations, and plays an important role in the learning environment. It is used to acquire a basic understanding of the architectures and the internal workings of components. Reverse engineering is often used to confirm and verify one's understanding of the system and to discover new ideas. In addition, reverse engineering a competitor's product is a common practice in acquiring another company's knowledge. In a sense, knowledge is encapsulated in the product, and the product becomes the medium for conveying knowledge.

Reverse engineering plays a critical role in the ongoing design and development of large-scale software-controlled systems. In this case, development primarily involves modifying the existing software, which requires designers to understand how the current system works. The primary method for this level of understanding is through studying the code itself — not studying the documents or comments. During their research on software program comprehension, Bennet and Morgan (1987, p. 13) cite the engineer, who explained: "If you understand the source code, you know what the program does, if you understand the comments you may or may not understand the program." When Cockburn (2000) asked software maintenance engineers how they manage to make program updates in the face of out-of-date documentation, they answered "that they 'just look around,' they don't trust the documentation in any case — they just read the code" (Cockburn, 2000, p. 11). Understanding complex software systems is a nontrivial task. Indeed, just finding the code that implements a particular function in a distributed processing system is often a challenging task. Consequently, an essential function of the learning environment is to provide reverse engineering tools that facilitate "bottom-up" understanding from the base system (e.g., code). One such tool, developed at Bell Laboratories, provides engineers with a mechanism for automatically structuring the code in multiple levels of detail from traces of software as the machine executes various functions (Klein, 1995). An engineer, for instance, could ask the system to execute a feature such as three-way calling, and the system would present the code in a hypertext format. The engineer could then "see what is happening" by studying the code at various levels of detail (Klein, 1995). This system is described in more detail in the section on electronic performance support systems (EPSS).

Summary

What is important to remember about knowledge management is that it is not just one initiative; it is the integration of many initiatives working together to ensure that knowledge is created and shared throughout organisations. How exactly this is done is unique to each organisation. Therefore, for the corporations, it is important to develop a shared vision of knowledge management and what it means for the organisation. For example, Xerox analysed the activities that organisations associate with knowledge management and divided them into 10 distinct areas. These domains are (Powers, 1999):

(1) Sharing knowledge and best practices; (2) instilling responsibility for knowledge sharing; (3) capturing and reusing past experiences; (4) embedding knowledge in products, services, and processes; (5) producing knowledge as a product; (6) driving knowledge generation for innovation; (7) mapping networks of experts; (8) building and mining customer knowledge bases; (9) understanding and measuring the value of knowledge; and (10) leveraging intellectual assets.

Documentation Management, Knowledge Portals, and Information Brokering

Documentation and information brokering are critical components of the corporate learning environment. Documentation is used to denote the internal generation of structured information, while information brokering is used to denote the search, retrieval, and presentation of external information to specific target audiences.

A fundamental requirement of the corporate learning environment is capturing internal corporate knowledge and making it available to work teams. Various technologies are available to support this goal that range from simpler technologies such as document management systems, to more complex technologies such as knowledge portals. A document management system typically makes use of a back-end database for storing and managing resources. Resources can be made available to a Web browser either by a "publishing" operation, in which the HTML resources are created by the document management system, or by converting the resources to HTML "on-the-fly." Notice that with both of these approaches the HTML is created by the document management system — information providers never have to create the HTML files using an authoring tool. For more detailed information and examples of document management systems see Chapter V of this book.

Knowledge portals (K Portals) are single-point-access software systems intended to provide easy and timely access to information and to support communities of knowledge workers who share common goals. K Portals give users a common interface and access point to all data inside and outside the corporation, making location, navigation, and retrieval of information easier. Users can access any information appropriate to their needs and privileges without having to know its location or format. With just one search engine and indexing scheme, a single question can be asked to find the information rather than travelling to several destinations to find an item.

A survey by Agency.com found that knowledge portals are having a tremendous effect on workforce efficiency as 36% of employees reported that their portal greatly impacts their ability to work productively. By using their portal, employees reported saving an average of 2.8 hours per week or around 7% of their time. Intranet and corporate portal satisfaction is directly influenced by having the right content, features, and design factors. Employees estimate that their productivity savings would double if their portal were improved so they could easily access all information necessary to perform their jobs (Kaplan, 2001).

Out-of-the-box and customised knowledge portals provide a range of function. To be effective, a knowledge portal should include provisions for security, an intuitive user

interface, a search engine, indexing/cataloging, document management, business intelligence tools, and personalisation, customisation, application, and data integration. More specifically, knowledge portals embed the following common functionalities (Mack, Ravin, & Byrd, 2001):

- *Capturing and gathering documents.* K Portals capture and gather internal or external documents and enables relevant documentation to be stored at a single-point that all users can access. Typically, corporate documents are stored in multiple places such as file systems on individual workstations, Web sites on network servers, and document management systems such as Lotus Notes. K Portals can automatically gather these documents via a process called crawling, which starts from a given URL or another specific address, and then automatically and recursively follows all the links in each document.
- *Document analysis.* Once the documents are gathered, the portal applies text analysis in order to extract textual features, which characterise the document so that their content is available for subsequent organisation, retrieval, and use. In addition to the textual features, documents are also analysed for their extrinsic features (also called meta-data features) that include information about the creation date, author, category assignment within a classification scheme, confidentiality, and so forth. Often, this meta-data information is gathered by the crawling process, and the content is represented in XML format, with the meta-data features encoded by XML tags.

  Document organisation. When the crawler has finished its gathering task, the documents are automatically organised into clusters. Rather than a flat space of clusters, some K Portals have powerful clustering engines that are capable of building hierarchical structures containing clusters and subclusters. Control points for such clustering engines include the critical size, the intracluster similarity metric, and the number of subclusters to build. Once the clusterer has finished its work, the clusters are named by inspecting the final cluster contents and choosing the best features to serve as names. Different K Portals utilise different approaches to categorise documents (see Baeza-Yates & Riberio-Neto, 1999). The major differences among the categorisation systems concern the types of features they use, the way in which they represent the features associated with categories, and the way in which they compare documents. Unfortunately, no one technique is entirely sufficient and all the methods need domain expertise and some degree of administrative skill.
- *Document search.* Once information is categorised, the portals allow users to conduct document search. Typical search engines provide a basic query function. More advanced search functions include: (1) prompted query refinement (PQR), which is a technique assisting the user in interactively refining the query, until a satisfactory set of focused and relevant information is returned; and (2) relevance feedback, in which the user finds at least one relevant document in the returned list, they submit this feedback to the engine and request to see more such documents.

Automatic question answering is another method where the user asks a full natural language question, natural language analysis determines the question focus, or the intended answer type, and brings up the response to that question. Some portal vendors such as Plumtree Software, offers an alternative to search system, in which personal information is utilised by the system to automatically generate searches on some attribute and present results to users (Plumtree Software, 2002). With this capability, users can be alerted about new documents related to their area of interest.

- *Browsing and navigation.* Since portals are built to assist users with large quantities of information, they need to embed an intuitive navigation system with built-in automatic summarisation tools that extract the most important information from documents and display it to the users. Four types of summarisation tools exist: (1) longer informative summaries (about 20 to 25% of the document length); (2) shorter indicative summaries (one to three sentences long); (3) query-based summaries (typically very short, involving the most important sentences where the query words are mentioned); and (4) keyword summaries that present a simple list of technical terms. Studies have shown that indicative summaries are sufficient for humans to complete tasks without having to read the entire document, thereby saving considerable time and effort (Hand, 1997).
- *Support for analysis, synthesis, and authoring of information.* Searching and browsing are a first step, but the information returned needs to be utilised for task purposes. Some more advanced knowledge portals are configured to also serve as application portals (enterprise application portals or EAPs), allowing the user to launch these applications within the portal and to interact with the data from the source. Also, some knowledge portals, such as Grapevine and the former Knowledge X, have the capability to generate relationship maps or graphic visualisation of entities and relationships that express organisational structures, connections among people, and project-related topics and artefacts. The goal of these tools is to provide a heterogeneous and open-ended workplace for representing objects and relationships that help users discover potential new relationships. Representations of entities and relations are integrated to some extent with databases containing information-describing entities, such as organisational, personnel, and project-related databases. Knowledge portals may also include links to resources and biographical information, bulletin boards, frequently accessed documents, highlighted news, and success stories.
- *Project collaboration.* While most knowledge portals leave the collaboration to third-party software, some K portals, like the ICM AssetWeb, exist in a workstation environment that includes tools for collaboration, such as electronic-mail, calendar, real-time meeting support with shared applications that are integrated with telephony, instant messaging. Also, K Portals might include collaborative authoring tools to allow multiple authors to keep track of multiple contributions, annotate contributions of coauthors, and merge multiple edits.
- *Links to resources and biographical information, bulletin boards, frequently accessed documents, highlighted news, and success stories.* Knowledge portals are also beginning to help organisations capture and leverage their intellectual assets by facilitating assembly of communities of interest, best practice, and expert systems within a single, intuitive, Web-based user interface. Knowledge portals should be viewed as an evolving technology platform, and in the future, knowledge portals may also incorporate streaming video and audio to include e-learning components, thereby potentially reducing overall organisational training costs.

Knowledge portals might be limiting when users want to go beyond accessing information and actually interacting with applications such as procurement, marketing, and supply chain management to improve and enhance productivity or customer service. Thus, portal vendors are poised to take the portal market to a new level that includes integrating applications into a unified portal interface that serves as a single focus for all applications and data accessed. Such portals should also be customisable for individual users, job functions, and corporate identities. ERP vendors such as Oracle and SAP have developed solutions that are tightly coupled with their enterprise applications. Unfortunately, their portals do not integrate well with competitive products, nor with applications beyond their product mix. These are critical limitations since most companies have best-of-breed strategies that require a vendor-neutral solution. To be an effective solution, the following characteristics are critical (Kao, 2001):

- Ability to integrate with any business application from any vendor.
- Ability to integrate with enterprise applications such as CRM, ERP, and legacy systems. Integration should be bi-directional, so users can create and modify data in underlying applications.
- It should support complex workflow to streamline business processes across different applications and make them work together seamlessly.
- It should run independent of the operating system, Web server, or Web browser.
- It should offer secure access that restricts available information and actions based on a user's role and privileges.

EPSS, Learning Management, and Training Management

Methods associated with knowledge management and knowledge portals indirectly support learning; the user decides the what, the when, and the how of engaging in some learning activity. Other learning methods are designed and prescribed by the organisation to meet predefined learning or performance objectives. Direct learning methods include training courses, self-paced learning modules, and online help and can be classified as electronic performance support, learning management, and training management.

Electronic Performance Support Systems (EPSS)

In many day-to-day job tasks, people learn on an as-needed basis while engaged in performing tasks through the use of job aids, procedural manuals, and online help. In the learning and performance community, these tools are referred to as performance support systems and when implemented online they are called electronic performance support systems (EPSS). In creating these systems, system analysts, instructional designers, or technical writers conduct detailed task analysis of standardised routine tasks and then create task-oriented procedural instructions and directions for performing each step.

While these systems are normally designed for routine tasks, recent performance support systems are now being designed to support the situations involving knowledge-intensive problem-solving, where individual tasks cannot be predetermined. These tools, instead of just "telling" people what to do, are designed to facilitate the development of mental models that enable operations staff and maintenance staff to "see" what is happening in the system. This, in turn, allows them to determine the tasks that need to be executed when something fails. Recent examples of this change include new displays in complex systems such as nuclear power plants that present graphical diagrams and animations of the operating system.

Another example of EPSS for the knowledge worker is software discovery tools. One tool mentioned was developed by Bell Labs for switching systems software developers (Klein, 1995). This tool automatically generates scenario-based documentation from traces of the code as the machine executes calls and features such as call waiting. Testing tools capture code as software executes and then puts the code into a file. Other tools interface these files to a hypertext system. In addition to generating hypertext, tools were created to animate the call flow by translating the trace files into a format that a graphical animation system could execute. The hypertext and animation systems use a simple ontology to describe devices: function, structure, and mechanism — how structure implements function (Weld, 1983). The top-level page lists the sequence of functional events in a call (e.g., off-hook, dialing, talking) and graphically depicts the message passing between processes in sequential order. An engineer can click on an event (e.g., collect digits) and go to a page containing top-level routines for that event. This page then shows the next level of detail including functional events and the corresponding code. This system enables engineers to obtain an accurate picture of how any feature works at anytime for any version of the software. As more people move to designing systems, the learning environment will need to focus on supporting knowledge-intensive tasks by making systems visible and inspectable (Brown & de Kleer, 1980).

Situation Oscilloscopes

In addition to supporting knowledge-intensive tasks, performance support systems need to be designed to support team performance. These tools are required to help coordinate tasks and to support shared situation awareness and mental models. For example, a camera attached to the doctor's head so that all the members of the surgical team can see what is happening greatly improves team performance over situations in which team members are just told what to do by the surgeon (Edmondson, 2003). Likewise,

communication tools that keep firefighters in contact with each other enables everyone involved in the emergency situation to have the same situation model of the incident as it unfolds (Klein, 1998). We have coined the term "situation oscilloscopes" to describe the set of tools to enable people to "see the invisible" and construct coherent pictures of complex situations. These tools would:

- Fuse information from multiple sources of data into a coherent picture (e.g.,enabling commanders in a battlefield situation to integrate multiple sources of intelligence and to understand what is happening across both local and global geographical areas).
- Enable a person to see and understand what is happening in complex systems and devices in which the basic components are invisible (e.g., the software code that is executed in switching systems or handheld wireless devices).
- Enable team members to know what is happening in situations in which multiple activities are being performed simultaneously by various members that are geographically dispersed (e.g., the firefighters, medical staff, and policemen engaged in dealing with an explosion).

Learning Management

The term "learning management" is used to denote the wide-scale use and interest in e-learning that has spread across most enterprises throughout the world. This form of instruction is delivered over the Web to individual learners as they need it. While many of the learning management systems offer integrated learning solutions that provide synchronous learning and virtual classrooms, we are using the term here to label the class of instructional activities associated with providing learning on-demand, at anytime, and at any place. In addition, learning is self-paced: Instructional materials provide all the learning activities, and often an instructor is not involved. Learning management systems focus on the delivery of small, self-sufficient modules of instruction rather than courses. These modules are often referred to as learning nuggets, learning bites, or some other appealing name, to indicate compactness and a narrowly defined chunk of content. Learning management systems are moving toward an open architecture, and when content conforms to standards such as SCORM, these modules are referred to as "learning objects" (Dillon & Hallett, 2001; Wiley, 2002). Learning management systems either prescribe a module as part of a performance-based or competency-based training program, or they enable the user to select modules when he or she wants to learn some particular piece of content.

The key attribute of this form of learning, when practiced as "learning on-demand," is that the user must supply the context for learning. The learner has a learning agenda and a reason for engaging in the instruction. In other words, the learner knows *what* they need to learn and *why* they need to learn it.

Training Management

Training management refers to the traditional form of group-paced, instructor-led training. Typical management functions include scheduling classes, finding instructors, registering students, and, when conducted in a physical location, finding classroom space. We show this form of training at the bottom of learning curve to stress that it is well suited for novices. Hereby, we must stress that expertise is highly contextual. A person can be an expert in one area but a novice in another. In our rapidly changing environment, people are constantly required to learn new technologies, processes, and tools. Therefore, this form of instruction now calls for development seminars and training focused on providing new information to experts as well as providing training to novices. The extent of this need is illustrated by the proliferation of vendors that provide public and corporate-tailored training courses and seminars to professionals. For example, TRA, a telecommunications training vendor, conducted over 1,500 seminars in 2001 that focused primarily on keeping experienced engineers up-to-date on the latest technology in their fields of expertise.

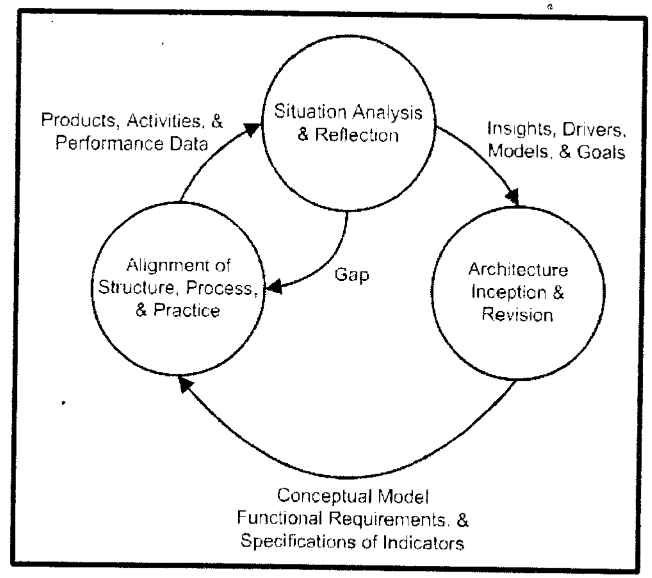
The distinguishing attribute of this form of instruction is that it must provide context for the learner. For novices, it must develop the reason behind the content rather than solely teaching the content. That is, the learner needs to learn what is important to know and why they need to know it. For experienced professionals, instruction must put the content in the context of what they already know — what is different, what is changing, and what is behind the changes. Experienced staff members often bring a very specific learning agenda with them to class. Consequently, conducting instruction requires teachers with both depth and breadth of content knowledge in order to present the broad framework, to relate content in historical perspective, and to answer a wide range of detailed technical questions of interest to individual students.

Developing the Corporate Learning Environment

The learning environment is fluid. It changes as the parent organisation changes. Consequently, the process for developing the learning environment should be iterative and adaptive. Our model (see Figure 3) organises the work involved in creating and sustaining a learning environment into three phases: (1) analysis and reflection, (2) architecture inception and revision, and (3) alignment. The model is depicted circular to illustrate the iterative nature of its processes.

The gist of the model might be best viewed in light of the following paradox; you should understand the situation before you try to change it, but, if you really want to understand the situation, try changing it. Continuous development incorporates the traditional roles of analysis, planning, and execution but realises that you cannot completely understand complex situations from analysis alone; that much of our understanding comes from feedback and reflection after executing a plan of action. Creating a viable learning environment that continuously adapts requires a process with four essential characteristics:

Figure 3. Adaptive process model



- Empowering people to change things as they see the need.
- Letting feedback drive the process.
- Focusing on incremental change.
- Having an architecture that enables incremental change.

The analysis phase incorporates "reflection" to emphasise that analysis deals as much with synthesis and learning as it does with decomposition — the outcomes of analysis are insights, revelations, and a better understanding of the situation. The intention of the architecture phase is to identify the key components of the learning environment. As Wenger (1998, p. 230) states: "It [architecture] is not a recipe; it does not tell a designer how to perform a specific design. But it does state what needs to be in place." In essence, the architecture shows how functionality is clumped and allocated across the components comprising the learning environment. The alignment phase involves the interventions, projects, and development efforts that may be undertaken in order to create the various aspects of the learning environment and to ensure that it supports the organisation's values, business goals, and core processes.

Situation Analysis and Reflection

The analysis phase consists of the common methods of analysis proposed by general systems theory such as cost-benefit analysis, front-end analysis, and job analysis. However, analysis must result in understanding and insight. When completing an analysis we must be able to say "I understand the situation." Analysis involves articulating values, reconciling dilemmas, understanding the communities of practice and the essential aspects of their work, knowing what is working and what is not, and knowing what adds value. IDEO, for instance, has well articulated values, understands how designers work, and knows what is fundamentally involved in doing good design work, knows that physical space and diverse teams add value to the design process, and it aligns their work processes and reward structure accordingly (Kelley, 2001). The traditional aspects of analysis, such as front-end analysis, are well documented elsewhere, therefore, we will not discuss them in detail. Rather, we will elaborate on cultural values and the processes that can be used for understanding how people learn and perform their work.

Cultural Values

There are two main reasons for clarifying values. One reason might be that the learning environment is dysfunctional because its processes are out of alignment with the true values of the organisation. The other reason is that corporations are becoming global networks of interacting people with different cultural values. Clarifying and reconciling values is critical to the company's success. Trompenaars and Hampden-Turner (1995) have outlined an effective process for dealing with cultural differences and incorporating values into effective business practices. They begin with the principle that different values need to be identified and reconciled rather than: (1) ignored, in which one insists on his or her values while the other is having to adapt to them; (2) one abandoning his or her values and adopting the other's values; or (3) being compromised, which often results in a lose-lose situation. Reconciliation is used to create solutions that fuse opposing views.

Trompenaars and Woolliams (2000) present an effective method for reconciling differences in underlying values. The process involves identifying dilemmas such as, "Does the organisation foster individual performance and creativity or is the focus on the larger group leading to cohesion and consensus?" They clarify a group's value by identifying where people would score on a continuum between the two extremes. Once the differences in values have been identified, the groups generate a solution that reconciles their differences. For example, in forming an alliance of the R&D activities between two companies in different countries, opposing views with regard to designing the reward structure were uncovered: One company based pay on individual performance, while the other based rewards on team performance. They reconciled their differences by installing a mixed system in which individuals were rewarded for team contribution and teams were rewarded for supporting individual excellence.

Understanding the Essential Aspects of Work

Two prevalent methods are used in understanding how work is performed. High-level methods such as surveys and focus groups are frequently used to determine what people perceive as "needs." These methods result in shallow understanding. All we know afterwards is "this is what the people say they need." A second method is at the other end of the analysis spectrum and involves conducting detailed job-task analysis and learning analysis. This method results in the specifications of the tasks involved in executing work. It is best used after learning components (e.g., user manuals, training programs, etc.) have been identified for development. Modeling is proposed as a "middle" method to help determine what components of the learning environment would add the most value.

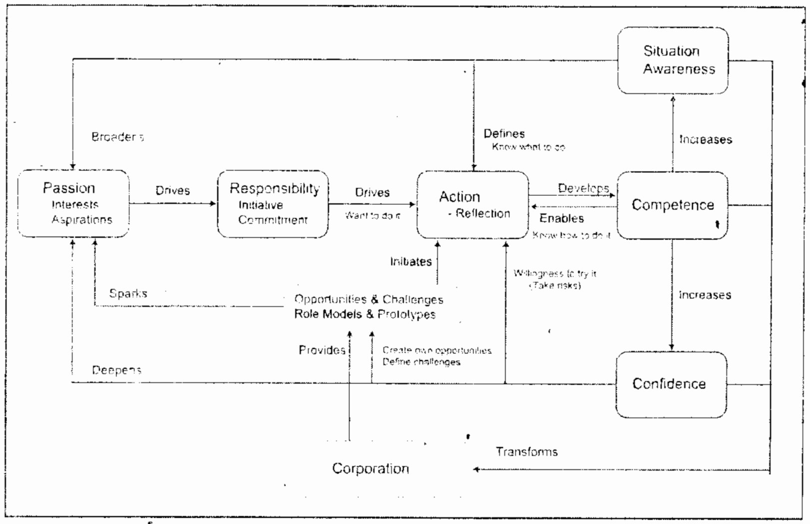
Models are representations of the real world that consolidate and articulate what we know and believe. Models can range in fidelity from simple lists of characteristics to working prototypes. A simple but powerful model is a metaphor. Metaphors and analogies map structure, function, and processes from one system to another and quickly convey essential concepts of the entity under study. For example, comparing the learning environment to a factory that manufactures knowledge generates a different meaning than comparing it to a farm that cultivates knowledge. In software development, a "software factory" would spawn a different set of processes than processes designed to support a "software studio." In addition to metaphors, useful modeling techniques in analysing the learning environment and representing human learning include lists, diagrams, and personas. Cockburn (2000, p. 17), for example, summarises his understanding of software developers as a list of characteristics:

1. People are communicating beings, doing it best face-to-face, in-person, with real-time questions and answers.
2. People have trouble acting consistently over time.
3. People are highly variable, varying from day-to-day and place-to-place.
4. People generally want to be good citizens and are good at looking around, taking initiative, and doing whatever is needed to get the project to work.

An example of modeling in the form of diagrams is illustrated by a generalised model of learning which we use to design learning architectures. The model in Figure 4 is based on our understanding that learning is essentially a process of acting and reflecting.

The model can be interpreted as follows: Engagement begins with passion and responsibility. As Owens (1998, p. 1) puts it: "Without passion nobody is interested. Without responsibility, nothing gets done." Passion and responsibility provide the motivation for action and reflection, which, in turn, develops competence. Competence enables a person to engage in more complex action and increases one's confidence, which, in turn, increases the willingness to try new actions. Confidence also deepens one's interests. Situation awareness helps people figure out what makes sense to do. It also broadens one's interests. The model represents a positive self-reinforcing loop that results in

Figure 4. Learning and development model



continuous learning. This model, fused with Cockburn's list of characteristics (e.g., people are good citizens), implies that corporations can create an effective learning environment by aligning work with people's interests, providing challenging tasks to spark passion, and then empowering them.

In this context, Cooper (1999) proposes personas, an interesting and useful form of modeling. Personas are hypothetical archetypes derived from studying people in real situations or imagining how a particular person would act in hypothetical situations (Cooper, 2003). The personas are given names such as "Chuck" or "Mary" and exemplify the various characteristics and actions of individuals. Personas are used both as a design tool and as a communication tool. For instance, both potential customers and engineers think in terms of: "What would Mary do in this situation?" or "Would Chuck understand this?" Personas represent a more detailed view of analysis than other modeling techniques. They are primarily used for software development projects, but they also can add a great deal of value in understanding user needs and guiding the design of the learning environment.

Architecture Inception and Revision

Architectures are conceptual models that depict how functionality is grouped and allocated across various elements such as tools, curricula, programs, and so on. These elements also include organisational clusters of people and their interactions. Consequently, designing the learning architecture is not the exclusive responsibility of the training department, human resource department, or information technology department. As Wenger argues (1998, p. 234), "[C]ommunities of practice are already involved in the

Figure 5. Hypothetical learning architecture

Communities of Practice	Business Processes	Learning Processes	
		Spontaneous	Designed
Functional Group 1 Functional Group 2 Functional Group [n] Cross Group Boundaries		Examples: Mentoring Coaching Collaborating Exploring ideas Discussing models	Examples: Training programs Learning modules User manuals Certification programs EPSS
Customers			
	Content Management		
Interaction	Space		
Communication Access (Portals)	Infrastructure		

design of their own learning because ultimately they will decide what they need to learn, what it takes to be a full participant."

Architectures are highly idiosyncratic and dependent on context. Learning architectures will vary greatly depending on the nature of the organisation and the people's views of the world; thus, a systems analyst and an anthropologist would more than likely conceptualise very different learning architectures for the same organisation. Therefore, it is imperative that multiple views be incorporated in generating the architecture. However, while there is a great deal of variance in learning architectures, the architecture should be based on: (1) the key drivers important to the parent organisation, (2) a firm understanding of how people learn, (3) processes that enhance corporate values, and (4) practices that leverage the strengths of the organisation and exploit technology.

A hypothetical learning architecture is shown in Figure 5 for illustrative purposes. This architecture depicts a layered approach to allocating functionality and emphasises a focus on meeting the needs of different communities of practice. Layering is a mechanism to decompose and distribute functionality in such a way that elements in one layer support elements in the next higher level, but a change in one layer does not require a change in another level. For instance, providing a new Web browser (access) should not require a change in the format of the content.

Customers are included as a community of practice to indicate that supporting relationships with this population is a key concern. "Cross group boundaries" are included to stress the importance of addressing this aspect of a corporate learning environment. The architecture also indicates our intentions that existing business processes and practices are primary components of the learning environment. The idea is that one should first look at existing business processes and practices to improve the learning environment. Two types of learning processes are depicted. The intention is not to provide a taxonomy of learning types but to indicate that there are two general learning processes at work: (1) the kinds of activities that are planned and systematically designed such as training programs, courses, user manuals, certification programs, and so on; and (2) the learning activities that naturally and spontaneously occur during the normal day-to-day work activities, which may include such daily activities as collaborating on a design, answering questions, explaining things to each other, and exploring new ideas over lunch.

Alignment

The alignment phase involves the various interventions and projects that are initiated in order to improve the learning environment. The term "alignment" is used rather than "development" to stress that not all solutions require a development effort. A solution could just be a matter of making minor modifications in the process in order to better synchronise customer training with product versions. Similarly, a solution could just reside on changing reward structures in order to better recognise team innovation; or it might involve a large-scale developmental effort to implement a XML-based content management system that enables marketing and training staff to more easily reuse engineering drawings in sales presentations and training programs.

When conducting projects, the work is organised and managed according to some systematic process and normally follows some form of systems planning. Larman (2001) presents a useful framework for structuring projects in which project activities are conceptualised along two dimensions: disciplines and phases. Table 1 presents a modified version of this framework to accommodate both traditional and incremental approaches to development. The framework also applies to both small-scale projects, where individuals may have responsibility for multiple disciplines; and to large-scale projects, in which there are divisions of labour and individuals who specialise in a specific discipline.

With respect to the project phases, inception includes forming a concept, which may be somewhat fuzzy, and creating the business cases, scope, and initial estimates. In the

Table 1. Project work management framework

Disciplines	Project Phases			
	Inception	Elaboration	Construction	Transition
Analysis and Modeling	XXXXX	XX	X	
Design	XX	XXXX	XX	X
Development	XX	XXX	XXXX	XXX
Evaluation	XX	XXXX	XXXX	X
Deployment	XXXX	X	X	XXXXX
Project Management	XX	XXXX	XX	XXXX
Environment	XXXX	XXXX	X	X

Note. X refers to corresponding workload.

elaboration phase, the concept is clarified and refined, the core architecture is developed, and critical features are designed and developed. The construction phase involves developing the remaining elements and preparing for deployment.

With respect to disciplines, "environment" pertains to the work involved in creating the tools, systems, and processes used to support all phases of the project. Design refers to those aspects of the project in which specific artefacts are described, planned, and specified. Development refers to the actual creation of the artefact. Evaluation has a prominent role throughout the project and involves measuring, assessing, and verifying artefacts, features, and processes. During the initial phases, evaluation could include creating test plans and formulating evaluation methods to be used during transition. Also, change management and version control are key elements of project management.

Summary

The development model should be viewed as a continuous and rapid process, in which analysis, architecture inception, and alignment occur in rapid iterations. In some cases, development efforts are initiated when feedback indicates that stable learning processes are not working as well as they should. In other cases, the processes are similar to incremental design practices: Problems are studied or emerging technology provides new opportunities; analysis is conducted and a conceptual solution emerges that requires modification to the learning architecture. Then, a project is initiated to implement the concept by creating new components or artefacts. After deployment, feedback provides new insights, and modifications are made to the architecture and elements.

Conclusion and Future Directions

The notion of a "learning environment" appears to provide a viable concept for synthesizing and making sense of the array of components, systems, and processes designed to support knowledge creation and movement, document management, e-learning, and performance support. Furthermore, the concept of a learning environment can serve as the unifying construct to guide the "convergence," "fusion," "blending," and "integration" of systems and functionality as prevalently discussed both in the research literature and in product descriptions of vendors. A learning environment with a well-defined architecture provides a viable mechanism for guiding the convergence of multiple systems and components into a seamless environment providing elegant access to content, multimedia learning modules, and collaborative workspaces. Standards for various artefacts, such as learning objects, modeling languages, content structures, and authoring tools, will enable the rapid construction of instructional materials, marketing presentations, and other elements that are designed to facilitate learning and knowledge flow.

A central conceptual trend is the movement toward a focus on people rather than process. This is indicated by the language and the metaphors currently being used to describe the rationale underlying knowledge management and learning management systems.

Phrases such as "knowledge flow" rather than "storing and retrieving knowledge," "cultivating relationships" rather than "transferring knowledge," and "appealing to higher order goals" rather than "providing incentives" indicate a movement away from the factory metaphor that has guided much of the past decision-making. One of the current challenges is to find new metaphors more in tune with describing the nature of the knowledge worker and the knowledge economy. We prefer to use "studio" as a metaphor for thinking about ways to organise work and design learning environments.

The next generation learning environments will reflect and accommodate the next generation corporate organisational structures, which may be profoundly different from today's organisation. The MIT Scenario Working Group has presented two possible scenarios for the 21st century which they label as "shifting networks of small firms" and "all encompassing virtual countries" (Laubacher & Malone, 1997). Scenario one, small companies-large networks, essentially represents radical outsourcing and consists of small firms linked by networks that come together for various projects and then dissolve when the work is completed. An example of this form of enterprise is the U.S. display division of Nokia, which achieved revenues of over $150 million in 1995 with only five employees and a constellation of small firms performing all the sales, technical support, logistics, and advertising tasks.

The virtual countries scenario involves massive global conglomerates with operations in almost every industry. In this case, a person might work for a company named Toyota-General Electric-Seimens. These companies would be owned by employees, and efficiency would be ensured by the use of organisational designers who "travel through the massive alliances, brokering partnerships, and helping make sure that people communicate effectively across boundaries" (Laubacher & Malone, 1997, p. 10).

These two scenarios present interesting challenges to the learning environment. First of all, designing a learning environment to support the large "virtual country" corporation is not the most critical problem to address because: (1) current processes are geared for large populations performing standardised tasks; (2) large firms can cost-justify providing the resources to develop learning programs and support professional development; and (3) we know how to "scale up." The essential problem is how to "scale down" in order to meet the learning needs of small fragmented populations. Even in our present situation, the fragmented groups within large corporations are a major inhibitor to the wide-scale adoption of e-learning (Dillon & Hallett, 2001).

We see three venues evolving to provide continuous learning opportunities for professional knowledge workers dispersed across small firms or functional groups within large corporations. First, small content vendors, such as TRA, operating in narrowly defined domains will provide seminars and courses designed to keep professional staff up-to-date on emerging technology. Second, professional guilds and associations will fund, support and sponsor the development of content in order to meet the professional development needs of their members (Laubacher & Malone, 1997). Third, universities will extend their mission beyond "certification" to include continuous development of professionals. Examples of this trend are MIT, which is making all their content available to alumni (MIT, 2001), and the University of South Florida's School of Engineering, which is extending their continuing education efforts to include the development and delivery of non-credit seminars for engineers (USF, 2004). We are exploring a distance learning

model to simultaneously provide credit and non-credit instruction. Courses in this model will consist of self-paced learning modules and virtual seminars. People wanting to learn a specific topic but not interested in earning credit would pay a small fee to attend the seminars and access the instructional modules.

Given the rapid movement toward standards for structuring content and the emerging technologies supporting virtual space, we see the learning environment of the future consisting of international networks of databases housing content objects adhering to standards, elegant access to the content, ubiquitous virtual spaces, and authoring tools that enable content vendors, guilds, and universities to rapidly construct, manage, and deliver a wide range of educational programs, professional seminars, and multimedia instruction.

References

Ambler, S.W. (2003). The principles of agile modeling. Retrieved May 26, 2004, from *http://www.agilemodeling.com/principles.htm*

Baeza-Yates, R., & Riberio-Neto, B. (Eds.) (1999). *Modern information retrieval*. New York: ACM Press.

Barron, T. (2000). A smarter Frankenstein: The merging of e-learning and knowledge management. Retrieved April 20, 2004, from *http://www.learningcircuits.org/aug2000/barron.html*

Bennet, R.W., & Morgan, T.C. (1987). *Comprehension of large programs in the context of a program maintenance task* (technical memorandum). Murray Hill, NJ: AT&T Bell Laboratories.

Brown, J.S., & de Kleer, J. (1980). Towards a theory of qualitative reasoning about mechanisms and its role in troubleshooting. In J. Rasmussen, & W. B. Rouse (Eds.), *Human detection and diagnosis of system failures*. New York: North-Holland.

Brown, J.S., & Duguid, P. (2000). Practice vs. process: The tension that won't go away. *Knowledge Directions*, 86-96.

Cockburn, A. (1997). Structuring use cases with goals. Retrieved March 3, 2004, from *http://alistair.cockburn.us/crystal/articles/sucwg/structuringucswithgoals.htm*

Cockburn, A. (2000). Characterizing people as non-linear, first-order components in software development. Paper presented at the fourth *International Multi-Conference on Systems, Cybernetics and Informatics*, Orlando, FL.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *IEEE Computer, 34*(11), 131-133.

Cooper, A. (1999). *The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity*. Indianapolis, IN: SAMS.

Cooper, A. (2003). The origin of personas. Retrieved April 15, 2004, from *http://www.cooper.com/content/insights/newsletters/2003_08/Origin_of_Personas.asp*

De Groot, A.D. (1965). *Thought and choice in chess*. The Hague, The Netherlands: Mouton.

Dillon, P., & Hallett, C. (2001). *Powering the leap to maturity: The eLearning ecosystem.* San Jose, CA: Cisco Systems, Inc.

Dreyfus, H.L., & Dreyfus, S.E. (1986). *Mind over machine: The power of human intuition and expertise in the era of the computer.* New York: The Free Press.

Drucker, P. (1993). *Post-capitalist society*. London: Butterworth Heinemann.

Edmondson, A.C. (2002). Managing the risk of learning: Psychological safety in work team. Retrieved January 5, 2004, from *http://www.hbs.edu/research/facpubs/workingpapers/papers2/0102/02-062.pdf*

Edmondson, A.C. (2003). Framing for learning: Lessons in successful technology implementation. *California Management Review, 45*(2), 34-54.

Edmondson, A.C., Winslow, A., Bohmer, R., & Pisano, G. (2003). Learning how and learning what: Effects of tacit and codified knowledge on performance improvement following technology adoption. *Decision Sciences, 34*(2), 197-223.

Eseryel, D., & Spector, J.M. (2000). Assessing adaptive instructional design tools and methods in ADAPT-IT. In M. Simonson (Ed.), *Annual Proceedings of Selected Research and Development Papers Presented at the National Convention of the Association for Educational Communications and Technology* (Vol. 1, pp. 121-129). Denver, CO: Association for Educational Communications and Technology.

Galamos, J.A. (1986). *Knowledge structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hand, T.F. (1997). A proposal for task based evaluation of text summarization systems. Paper presented at the *Conference on Intelligent Scalable Text Summarization*, Tampa, FL.

Hargadon, A., & Sutton, R.I. (1997). Technology brokering and innovation in a product development firm. *Administrative Science Quarterly, 42*, 716-749.

Jeffries, R. (2001). Natural XP: Documentation. Retrieved April 30, 2004, from *http://xprogramming.com/xpmag/natural.htm*

Kao, C. (2001, February). Enterprise application portals. Retrieved February 17, 2004, from *http://www.bijonline.com/Article.asp?ArticleID=305*

Kaplan, M. (2001). Intranets and corporate portals: User study. Retrieved February 27, 2004, from *http://knowledgemanagement.ittoolbox.com/browse.asp?c=KMPeerPublishing&r=%2Fpub%2FTR031802.pdf*

Kelley, T. (2001). *The art of innovation: Lessons in creativity from IDEO, America's leading design firm.* New York: Doubleday.

Kelly, R., & Caplan, J. (1993). How Bell Labs creates star performers. *Harvard Business Review, 71*(4), 128-139.

Kerievsky, J. (2001). Continuous learning. Retrieved May 15, 2004, from *http://www.industriallogic.com/xp/ContinuousLearning.pdf*

Kintsch, W. (1986). Learning from text. *Cognition and Instruction, 3*, 87-108.

Klein, G. (1998). *Sources of power: How people make decisions*. Cambridge, MA: MIT Press.

Klein, J. (1995). *Implications of emerging technologies and business practices to technical training and instructional systems development.* Paper presented at the Instructional Systems Academic Forum, Florida State University.

Klinger, D. (2003). *Handbook of team cognitive task analysis*. Fairborn, OH: Klein Associates, Inc.

Kuipers, B., & Kassirer, J.P. (1984). Causal reasoning in medicine: Analysis of a protocol. *Cognitive Science, 8*, 363-385.

Larman, C. (2001). *Applying UML and patterns: An introduction to object-oriented analysis and design and the unified process* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Laubacher, R.J., & Malone, T.W. (1997). *Two scenarios for 21st century organizations: Shifting networks of small firms or all-encompassing virtual countries?* (no. 21C WP #001). Boston: Massachusetts Institute of Technology.

Leonard-Barton, D. (1992). Core capabilities and core rigidities: A paradox in managing new product development. *Strategic Management Journal, 13*(5), 363-380.

Mack, R., Ravin, Y., & Byrd, R.J. (2001). Knowledge portals and the emerging digital knowledge workplace. *IBM Systems Journal, 40*(4), 925-955.

Manasco, B. (1999). Leading lights: Technology designer Michael Schrage. Retrieved March 13, 2004, from *http://www.webcom.com/quantera/schrage.html*

Miller, E. (2003). Digital manufacturing: Expanding the scope of product life cycle management. Retrieved May 1, 2004, from *http://www.cimdata.com/publications/article03-02_digital_man.html*

MIT. (2001). MIT to make nearly all course materials available free on the World Wide Web. Retrieved May 20, 2004, from *http://web.mit.edu/newsoffice/2001/ocw.html*

Nelson, R. (1991). Why do firms differ, and how does it matter? *Strategic Management Journal, 12*, 61-74.

Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. New York: Oxford University Press.

Owens, H. (1998). Emerging Order in Open Space. Retrieved May 26, 2004, from *http://www.openspaceworld.org./wiki/wiki/wiki.cgi?EmergingOrderInOpenSpace*

Plumtree Software, I. (2002). Search in the Plumtree corporate portal. Retrieved February 13, 2004, from *http://reg.plumtree.com/eMarketing/start.swe?SWECmd=InvokeMethod&SWEMethod=ShowWebOffer&SWEView=URLRouter3&SWEApplet=OfferRoute&SWENargs=OID=1-39H7M,CID=1-2MENU9,COID=1-2MENUD,CON=1-389KHK#WhitePapers*

Polanyi, M. (1966). *The tacit dimension*. London: Routledge & Kegan Paul.

Powers, V.J. (1999). Xerox creates a knowledge-sharing culture through grassroots efforts. Retrieved April 13, 2004, from *http://www.askmecorp.com/pdf/Xerox.pdf*

Prahalad, C.K., & Hamel, G. (1990). The core competencies of the corporation. *Harvard Business Review, 68*(3), 79-91.

Rasmussen, J. (1986). *Information processing and human-machine interaction.* New York: North Holland.

Reich, R.B. (1998). The company of the future. *Fast Company, 19*, 124-152.

Rouse, W.B., & Morris, N.M. (1985). *On looking into the black box: Prospects and limits in search of mental models* (no. 85-2). Atlanta, GA: Georgia Institute of Technology, Center for Man-Machine Systems Research.

Senge, P.M. (1990). *The fifth discipline*. New York: Doubleday/Currency.

Soloway, E. (1986). Learning to program = learning to construct mechanisms and explanations. *Communications of the ACM, 29*(9), 850-858.

Thomke, S., & von Hipple, E. (2002). Customers as innovators. *Harvard Business Review, 4*, 5-11.

Trompenaars, F., & Hampden-Turner, C. (1995). *The seven cultures of capitalism: Value systems for creating wealth in Britain, the United States, Germany, France, Japan, Sweden and the Netherlands*. London: Judy Piatkus Publishers Ltd.

Trompenaars, F., & Woolliams, P. (2000). When two worlds collide. Retrieved May 20, 2004, from *http://www.7d-culture.nl/Downloads/WhenTwoWorldsCollide.pdf*

USF. (2004). Florida engineering education delivery system (FEEDS). Retrieved April 30, 2004, from *http://feeds.eng.usf.edu/*

von Krogh, G., Ichijo, K., & Nonaka, I. (2000). *Enabling knowledge creation*. New York: Oxford University Press.

Weidenhaupt, K., Pohl, K., Jarke, M., & Haumer, P. (1998). Scenarios in system development: Current practice. *IEEE Software, 15*, 34-45.

Weld, D. S. (1983). *Explaining complex engineered devices* (no. 5489). Cambridge, MA: BBN.

Wenger, E. (1998). *Communities of practice*. Cambridge, UK: Cambridge University Press.

Wiley, D. (2002). *Instructional use of learning objects*. Bloomington, IN: AECT.

Wittgenstein, L. (1958). *Philosophical investigations*. New York: Macmillan.

Internet Session:

Formulating a Learning Environment Development Process Based on Agile Methods

http://www.agilemodeling.com/ http://www2.umassd.edu/SWPI/xp/papers.html

Interaction:

Review the information presented in the Web sites on agile development methods and philosophy. Then, prepare a brief presentation on (1) how the core concepts can be used in designing the overall learning environment *or* (2) the impact on the learning environment if a large software development corporation decided to start using agile development processes instead of the traditional systems approach.

Case Study

Learning Environment for a New High-Tech Company

An established high-tech company has decided to expand into the wireless market by producing handheld devices. An example product could be handheld scanners that people in a store would use to take inventory — they scan each product on a shelf and the data is relayed over a wireless network to a computer system. An entirely new subsidiary, Portable Wireless Inc. (PWI), is being formed. It will have its own research, engineering, manufacturing, marketing, and sales staff. PWI is quickly being established as a viable operating company by buying smaller firms that make similar devices or have relevant technology. As of this date, they have bought six corporations, and PWI now has manufacturing plants in Syracuse, NY, and Rome Italy, where two of the companies they bought are located. They have a research and development department in Syracuse, NY, and the Syracuse facilities will serve as corporate headquarters. They also have research, development, and engineering departments in Rome, Tampa, Florida, and Singapore. These departments were part of the corporations bought by PWI. A company they bought that was particularly strong in marketing is located in Montreal, Canada, and it will assume primary responsibility for marketing and sales products.

Questions:

1. What do you see as the critical issues to address in establishing a viable learning environment for PWI?
2. What development strategy and process would you initially use to create the learning architecture and then sustain and improve the learning environment over time? That is, how would you go about creating the learning environment?
3. Assuming that you have the full support of upper-level management and reasonable resources, what would be your design for a state-of-the-art learning environment for PWI? That is, what would be your learning architecture?

Useful URLs

Agile Alliance: *http://www.agilealliance.org/home*

A collection of articles on iterative/incremental design methods: *http://www2.umassd.edu/SWPI/xp/papers.html*

Community Intelligence Labs: *http://www.co-i-l.com/coil/index.shtml*

The Knowledge Management Resource Center: *http://www.kmresource.com/sources.htm*

Trompenaars Hampden-Turner: *http://www.thtconsulting.com/index1.html*

Mental Models introduction: *http://www.tcd.ie/Psychology/Ruth_Byrne/mental_models/*

Situation Awareness: *http://www.satechnologies.com/html/papers/teamsa.shtml*

Situation Awareness: *http://www.thoughtlink.com/wae.htm*

Learning Tech Navigator: *http://www.ltnavigator.com/index.htm*

State-of-the-Art Knowledge Portal product: *http://www.entopia.com/index.html*

State-of-the-Art EPSS product: *http://www.knoa.com/gui/index.asp*

Learning Systems Architecture Lab: *http://www.lsal.cmu.edu/lsal/index.html*

Example learning systems architecture product: *http://www.trifus.com/*

Further Readings

Dillon, P. & Hallett, C. (2001, October). *Powering the leap to maturity: The eLearning ecosystem.* Cisco Systems white paper.

Edmondson, A. C. (2002). Managing the risk of learning: Psychological safety in work team. Available at *http://www.hbs.edu/research/facpubs/workingpapers/papers2/0102/02-062.pdf*

Kelley, T. (2001). *The art of innovation: Lessons in creativity from IDEO, America's Leading Design Firm.* New York: Doubleday.

Klein, G. (1998). *Sources of power: How people make decisions.* Cambridge, MA: MIT Press.

Senge, P.M. (1990). *The fifth discipline.* New York: Doubleday/Currency.

Thomke, S., & von Hipple, E. (2002). Customers as innovators. *Harvard Business Review, 4,* 5-11.

Trompenaars, F. & Hampden-Turner, C. (1995) *The seven cultures of capitalism: Value systems for creating wealth in Britain, the United States, Germany, France, Japan, Sweden and the Netherlands.* London: Judy Piatkus Publishers Ltd.

Wenger, E. (1998). *Communities of practice.* Cambridge, UK: Cambridge University Press.

Possible Paper Titles/Essays

How Reconciling Values Enhances the Corporate Learning Environment

The Use of Simulations, Models, and Scenarios to Provide Effective Training

An Agile Instructional Systems Design Model

A Comparison of Knowledge Management Systems, Learning Management Systems, and Knowledge Portals

An Effective Learning Environment for Small, High-Tech Companies

Chapter II

Enabling Technologies for the Semantic Web

Kevin R. Parker, Idaho State University, USA

Editors' Notes

Kevin has a clear mission: to present the Semantic Web notion for everyone. While several of the aspects of the Semantic Web will be explained further in other chapters, readers unfamiliar with the Semantic Web issues should start thinking of the importance of the Semantic Web as an enabler of more effective intelligent knowledge and learning infrastructures. We have placed this chapter after the corporate learning environment chapter for obvious reasons. We want to converge two pillars of critical importance: on the one hand, leading and state-of-the-art research on theoretical foundations of the next generation knowledge and learning management; and on the other hand, leading edge technologies as those of Semantic Web.

This objective is present in every chapter of the book. We want theories and technologies to be applied in specific contexts toward the development of socio-technical systems aiming to provide a performance driven by knowledge and learning.

It is again worthy to mention, our involvement in the Special Interest Group on Semantic Web and Information Systems of the Association for Information Systems (*http://www.sigsemis.org*). We encourage you to visit our portal and consider becoming part of this community. An excellent point of reference for issues related to the Semantic Web is the AIS SIGSEMIS Bulletin, the official quarterly newsletter of the AIS SIGSEMIS, where research papers, research center presentations, and interviews of the leaders of SW provide important knowledge for the field. Moreover, the International Journal on Semantic Web and Information Systems published by IDEA Group Publishing, *http://www.idea-group.com*, sponsored by AIS SIGSEMIS, provides leading edge research outcomes. It provides an excellent addition to your portfolio of scientific journals.