Knowledge Management in Non-Collocated Environments: A Look at Centralized Vs. Distributed Design Approaches

Kevin C. Desouza, Anuradha Jayaraman, J. Roberto Evaristo
Department of Information & Decision Sciences, University of Illinois at Chicago
E-mail [kdesou1, ajayar1, evaristo]@uic.edu

Abstract

In the domain of project management, types of knowledge can be characterized as "knowledge in projects", "knowledge about projects", and "knowledge from projects". The use of information technology in the realm of knowledge management has been approached from two main angles: codification and personalization. In the following paper, we look at the implications that each technological approach has on the aggregation, transfer, and sensemaking of knowledge in distributed environments. Drawing on the strengths and limitations of each technique, we propose a hybrid model. Research questions pertaining to the management of knowledge systems are presented.

1. Introduction

The trend towards concentration on one's core competencies while striving for globalization, has transformed organizations from the classical hierarchical structures to decentralized and distributed topologies, to enable agility and flexibility. One prominent outcome of such a transformation is the concept of projects [3,14]. Turner [19, p.8] defines a project as "an endeavor in which human, material and financial resources are organized in a novel way, to undertake a unique scope of work, for a given specification, within constraints of cost and time, so as to achieve beneficial changes defined by quantitative and qualitative objectives". Projects have moved from being simple phenomena to manage i.e. single projects in a single location, to more complex entities which span geographical locations, multiple occurrences, and different organizational affiliations [10,14], information technology being the key enabler for the transformation. In the realm of project management, much of the efforts in incorporating technology have been to foster ubiquitous communication between project members while also enabling for ease of knowledge exchange [7].

Most efforts in the realm of knowledge management can be characterized as being either personalization or codification [12]. In the codification strategy, individual knowledge is amalgamated, put in a cohesive context, and made centrally available to members of the organization via databases and data warehouses. The codification strategy uses a document-to-person approach on the premise that knowledge can be effectively extracted and codified. Knowledge management using this approach is highly structured as compared to the personalization approach that is semi-structured.

The personalization approach does not impose a distinction between the knowledge and the knowledge provider. It recognizes the tacit dimension of knowledge and assumes that knowledge is shared mainly through direct person-to-person contacts. The role of information technology here is to facilitate communication between members of the organization through tools such as e-mail, group support systems, etc.

On closer examination of the above mentioned approaches, we can draw parallels to two popular models of computing, namely client-server and peer-to-peer. The client-server paradigm, wherein there is a centrally located resource used by multiple clients to request services for accomplishment of a task is common in most of the distributed computing environments. Peer-to-peer computing is a rather recent computing paradigm in which all nodes can take the role of either a clients or servers [16]. They can request information from any other node or peer on the network and can also serve content. Due to the centralization of the main resource provider, client-server computing is very similar to the codification strategy, and the distributed nature of peer-to-peer in which each node owns its resources and makes it available to the network can be seen as a parallel to the personalization strategy.

In this paper, we explore the issue of managing knowledge in distributed or non-collocated environments through the technological perspective. We look at the strengths and weaknesses of the centralized and decentralized approaches in the context of facilitation of effective and efficient aggregation, transfer and sensemaking, of knowledge in, about, and from projects. Following Katzy et al [14] we assert that knowledge management has direct bearings on project outcomes. The rest of the paper is organized as follows. We begin with a discussion on the various types of knowledge generated in projects, followed by a brief review of the typology of projects. Next, our research framework for the role of knowledge management systems is presented, along with a scrutiny of each approach. A hybrid approach is then proposed, followed by a discussion of research insights and practitioner implications.

2. Types of Knowledge

Knowledge generated by projects can be categorized under knowledge in projects, knowledge about projects, and knowledge from projects [3]. These three categories call for distinct roles played by information technology to enable effective and efficient knowledge management. We now discuss each type of knowledge.

2.1 Knowledge in projects

Knowledge in projects calls for a microscopic look at intimate insights generated within each individual project. Items of interest in this category include project schedules, milestones, minutes of meetings, training manuals, etc. Individual project members need to know when, what, how, where, and why something is being done and by whom, with the goal being to promote efficient and effective coordination of activities [14]. The more exchange of relevant knowledge that takes place the better off will be the team performance in an organization, as each member can learn from the other's experience and complement each other's endeavors. The use of group support systems and e-mail are common to enable sharing of knowledge in projects [5,6].

2.2 Knowledge about projects

From the macro perspective, an organization must have an inventory of all projects underway at any given time. This aids in planning and control of resources to maximize utilities. Knowledge of interest includes employee assignment to projects, return on investments, cost and benefit analysis, deadlines, customer commitments and expectations, etc. It is common for such knowledge to be generated at regular intervals such as weekly, monthly, or bi-monthly reports. Executive information systems and other reporting mechanisms are commonly used to generate knowledge about projects, perform aggregation and summarization, and finally present it in a cohesive manner [4,7].

2.3 Knowledge from projects

Knowledge from projects is a *post hoc* analysis and audit of key insights generated from carrying out projects. This knowledge is a key determinant of future project successes as one can learn how to function more optimally while avoiding mistakes committed in the past. Moreover, a company must preserve knowledge and experiences about past projects in order to successfully deploy future initiatives. In today's economy, employee turnover is high along with a tendency for opting for early retirement. As such, an organization must make every effort to capture personal knowledge to prevent loss of valuable know-how. To efficiently manage such knowledge one must enable its availability and transferability both across other projects being conducted at the same time and to future projects [14]. Knowledge sharing and intellectual capital transfer that takes place in a group setting and via formation of project teams is fostered through a knowledge management system. A vast majority of such systems employ the codification approach, in which a central repository holds knowledge under categories such as project reports, programming bugs, quality control reports, new developments, etc [5,6]. An important consideration while contributing *post hoc* knowledge from a project is to capture the context in which the knowledge was generated [6,14], because failure to do so will lower the real value of knowledge.

2.4 Knowledge management and project management

As pointed out by Katzy et al [14] there are several key distinctions between the notions of knowledge management and project management. Firstly, project management is a finite endeavor while an organization must manage knowledge in continuum. Secondly, project management is goal oriented and occurs in spite of differences in culture at the level of project members, organizations, or national entities. Knowledge management, however, is not an end in itself.

In spite of these distinctions, one can draw relations between the two concepts. Knowledge is used for goal attainment in projects, and projects generate knowledge for the organization. Organizations generate and dissipate vast amount of information and knowledge on a daily basis [17]. Agents or objects generate knowledge at the individual level in an organization. Drawing on Grant's [11] knowledge-based theory of the firm, knowledge creation lies at the individual level and the main role of the organization is to aggregate, integrate and exploit it in an optimal form. However, to be successful in generation of new knowledge one must tap into past experiences, know-how, and identify sources of knowledge. Hence, rather than independent concepts, project management and knowledge management are heavily interdependent.

3. Typology of Projects

Evaristo and Fenema [10] proposed a typology of projects, based on two dimensions: the number of locations (single versus multiple) and the number of projects (single versus multiple). The traditional scheme being a single project in a single location, we can also have multiple projects running at one location (co-located program). A distributed project is a single endeavor being conducted by individuals from multiple locations. Finally, the most complicated scenario is multiple projects conducted at multiple locations. Complexities can be attributed to managing multiple interdependencies across time, space, and projects [9,10]. Katzy et al [14] built on this framework to incorporate projects spanning multiple organizations. Adding the concept of affiliation dispersion, they characterized projects by low or high dispersion in geography and team member affiliation. Traditional projects are low in both dimensions, followed by distributed projects in which team members are geographically dispersed but have a common affiliation. Traditional interorganizational projects have low geographical dispersion but participating team members have high affiliations to their organizations [9]. In virtual projects, one finds high geographic and affiliation dispersion.

For the purpose of our research, we are concerned with organizations executing single or multiple projects that span single or multiple sites and involve single or multiple stakeholders. Considering the three dimensions of a project as number of sites, number of projects being executed and number of stake holders, we have eight different kinds of projects as indicated in Table 1. For our purpose, we are interested in non-collocated projects, leading to four different project scenarios, as indicated below. We assert that each dimension of a project i.e. number of sites, number of projects being executed and number of stakeholders, impose unique requirements on the type of technology infrastructure needed for knowledge management. For instance, the number of stakeholders raises the issues of technology compatibility, heterogeneity of groups, and security, while the notion of geographic dispersion calls for a discussion on the economics of dynamic availability of tools and access mechanisms. For example, consider an employee working at the company headquarters who accesses the knowledge base using a T1 connection, while his team member accesses the knowledge base using a modem. This difference in the economics such as bandwidth available calls for conscious system design considerations to satisfy heterogeneous groups. When managing multiple projects spread across multiple locations, the need for efficient and effective coordination is amplified [13].

Central to all projects is the issue of coordinating actions of members and executing tasks in an optimal manner. To meet this end, as mentioned previously, knowledge in projects must be aggregated, and integrated with knowledge about and from projects, and be readily transferable across time, space, and projects. White and Fortune [20] in their survey of over 236 project managers found clear communication channels, effective monitoring and feedback, taking account of past experiences, having access to relevant talent, and contextual awareness to be critical determinants of a project's outcome. Based on these and other factors, we now propose our theoretical model for scrutinizing technologies that foster such efforts.

Table 1: Typology of Projects

Number of Sites	Number of Projects	Number of Stakeholders
Single Location (Collocated)	Single	Single
		Multiple
	Multiple	Single
		Multiple
Multiple Locations (Non-Collocated)	Single	Single
		Multiple
	Multiple	Single
		Multiple

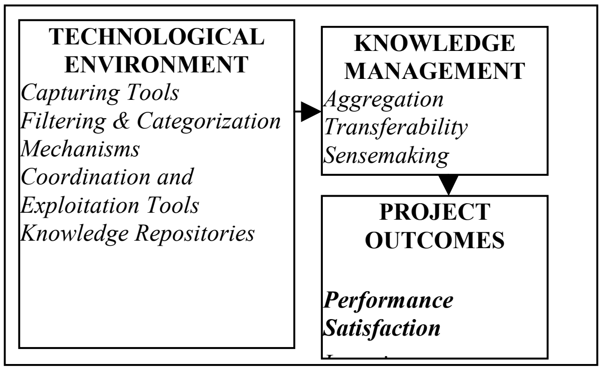
4. Theoretical Framework

Katzy et al [14] proposed a framework for developing a research agenda looking into knowledge management in projects. Their framework looked at the role played by project and team member characteristics, in addition to the external and technological environments, on knowledge management in projects. More specifically, knowledge management in projects was restricted to three actions: aggregation, transferability, and sensemaking. Our proposed framework can be viewed as a subset and focused adaptation of their model. We restrict ourselves to the effect of the technological environment on knowledge management. Here, technological environment is defined as the collection of design factors of the infrastructure to facilitate knowledge management. It includes items such as knowledge capturing mechanisms, coordination tools, filtering and categorization tools, and repositories. We first discuss various components of knowledge management systems and their effect on knowledge management, followed by a brief discussion about knowledge management and project management outcomes.

4.1 Knowledge capturing tools

Knowledge, for the most part, resides in the minds of employees in the form of know-how. Much of the knowledge residing with employees is in tacit form. To enable sharing across the organization, this knowledge needs to be transferred to an explicit format. According to Nonaka and Takeuchi [15], for tacit knowledge to be made explicit there is heavy reliance on figurative language and symbolism. Employees are more likely to communicate freely in an informal atmosphere with peers than when mandated by management [5]. Organizations can foster knowledge capture by recognizing contributions and providing incentives for such employees and motivating them to add to the knowledge base. Knowledge capturing tools are hence essential starting points to the process.

Figure 1: Theoretical Framework



Knowledge capturing tools should have certain capabilities to promote their use. Firstly, they should be perceived as being easy to use. Overly complicated knowledge capturing tools will deter employees from sharing their knowledge due to the high costs involved in inputting insights into the system. Secondly, these tools must allow for flexibility [2]. This factor is closely related to the ease of use. No two employees work the same way and hence the chances that two employees will record knowledge in a similar fashion are low. Being the front end of most knowledge management systems, the capturing module needs to allow for customization rather than force users into prescribed formats. Finally, such tools must be available real time. Since knowledge generation happens on a sporadic and unplanned basis, insights not captured immediately are lost.

4.2 Filtering and Categorizing Mechanisms

Once captured, knowledge needs to be filtered for relevance and categorized for ease of retrieval to enable for efficient and effective exploitation. Relevance of knowledge is a highly volatile concept as it is based entirely on the context of its production and usage. Chattaraj et al [2] frame the notion of relevance and the significance of categorization using the concept of *Knowledge Acceptance Behavior*. It is defined as the perception and acceptance of knowledge by the receiving unit (or agent) in an organization. Independent of the quality or quantity, for a function to truly exploit the nugget of insight, knowledge received must be in agreement with the functions' need. Otherwise, knowledge becomes a burden for the function. If the receiving functions don't perceive and accept any knowledge input as workable, or worthy to process, a particular input won't have any value in the system. However this sense of perception is highly volatile, as the specifications of workability, or worthiness, change in a changing business environment. Hence, there is need to categorize knowledge effectively so as to provide relevant sources of insight rather than burden team members by having them sift through large masses of irrelevant knowledge.

Common approaches to categorizing include sorting or arranging knowledge contents based on topic, provider, or date. Ranking mechanisms are in place to enable users to search for documents based on relevance, such as those found in common search engines such as Google. Consider the following: according to a study by Gartner Group, the amount of data collected by an organization doubles every year. Knowledge workers analyze only 5% of this data. Knowledge workers spend 60% of their time searching for important relationships in the data, 20% is spent on analyzing the discovered relationships, and only 10% is spent on applying the results of the analysis i.e. making decisions, implementing strategies and plans, etc. Moreover, Gartner group states that information overload reduces decision making capability by 50%. Information overload can be characterized as a case in which large directories of codified knowledge may never be used due to the overwhelming size.

To curtail the issue of information overload efficient and effective filtering and categorizing mechanisms need to be incorporated. There are two overriding desirable features for such tools. First, filtering should be conducted at two levels: global and local. Global filters are predefined and agreed upon thresholds for parsing knowledge contents by members of the organization. These filtering tools should be run at predetermined intervals and should not be subject to modification by individual agents in the organization. However, each agent or team member needs to be able to parse documents according to his or her customized filtering scheme. Hence, at the local level, customization of thresholds and parameters must be enabled. Secondly, in defining categories for knowledge storage one must be careful in considering the context. Context can be defined as the interrelated conditions in which something exists or occurs. Context can be looked at from two levels: global and local. While knowledge from one project can be aggregated with other similar projects into high level global categories, organizing knowledge in projects depends heavily on the localized context of the project and hence each project may have its own internal representation scheme.

4.3 Coordination and Exploitation tools

Coordination means joint involvement of organizational members in some activity involving information sharing, mutual goal setting, and decision making. KPMG's survey suggests that the biggest barrier to organizational knowledge usage is a blocked channel between knowledge provider and seeker. Blockage arises from factors such as temporal location, and lack of incentives for knowledge sharing [18]. Ruggles [18] study of 431 US and European companies show that 'creating a network of knowledge workers' and 'mapping internal knowledge' are the two top missions for effective knowledge management. Knowledge generated in projects is, for the most part, needed for coordination of activities among team members. Such knowledge as discussed before consists of project plans, schedules, status reports on each task or member, etc. It must hence be enabled for ease of transfer and retrieval by participating members. Moreover, such knowledge must portray real-time status of the project, as failure to update the knowledge renders it useless.

Exchange of explicit knowledge is relatively easy for electronic communities through e-mail or document exchange protocols. However exchange of tacit knowledge is easier when we have a shared context, co-location, and common language (verbal or non verbal cues) as it enables high levels of understanding among organizational members [1,5,6,8,9]. Nonaka and Takeuchi [15] identify the processes of socialization and externalization as means of transferring tacit knowledge. Socialization keeps the knowledge tacit during the transfer whereas externalization changes the tacit knowledge into explicit knowledge. Example of socialization includes on the job training and apprenticeships. Externalization includes the use of metaphors and analogies to trigger dialogue among individuals and to provide a mode for articulation of tacit insights. Some of the knowledge is, however, lost in the transfer.

Facilitating coordination requires clear identification of the sources of knowledge in and about projects. This calls for efficient and effective search mechanisms to reduce access times when trying to locate the source of knowledge. Proper coordination mechanisms need to be in place to facilitate easy access to knowledge repositories for their exploitation. Today almost all knowledge repositories are being web enabled to provide for the widest dissemination via the Internet or Intranets. To foster tacit knowledge sharing, organizations should allow for video and desktop conferencing as viable alternatives for knowledge dissemination. Tacit knowledge is experiential and hence one way to transfer such knowledge is to see how the knowledge is transformed into action [5,15]. Much like knowledge capturing tools they need to be easy to use, flexible to setup ad hoc knowledge exchange forums, and real time. This is crucial because a fluid ex ante nature of objectives, outcomes, and processes characterizes knowledge management systems. Within the knowledge management context, it is difficult to know a priori what information will be requested, who will request the information, who will supply the information, and when and how the information will be used.

4.4 Knowledge Repositories

Several key issues surface when addressing knowledge repositories. Firstly, how much and what knowledge to store is a common concern in the design of knowledge management systems. On the one hand, over filling knowledge repositories is a common blunder committed by many organizations [7]. Over filling knowledge repositories leads to larger search and retrieval times along with the retrieval of outdated and irrelevant information. Maintenance costs also increase as the size of a knowledge repository grows. Knowledge repositories thus have to be purged in a timely fashion to take out the old and input the new. The flip side is too little knowledge being stored in the repositories, which is also not good as it serves minimal purpose. Hence one has to strike a balance between under and over filling repositories. Secondly, an issue of interest is where the knowledge repository is located. Do we have one centralized repository? Or do we have multiple distributed repositories? If we opt for multiple repositories what criteria is used to decided what each repository will hold and how are they going to be linked or synchronized? Finally, in distributed environments, the issue of knowledge ownership needs to be addressed. In interorganizational projects who owns the knowledge generated in projects?

4.5 Knowledge Management Outcomes

The above factors make up the technical environment for developing knowledge management systems. All factors have associated effects on knowledge management. For our purposes we restrict knowledge management to the acts of aggregation, transferability, and sensemaking. Aggregation refers to the task of assimilating knowledge generated by individual units and putting it in a cohesive context for use by a wider audience. Here, individual units refer to members of a team in the context of knowledge in projects, and to single projects when discussing knowledge from and about projects. Transferability is the logistic act of moving knowledge from provider to consumer. We focus on three types of transfer. Firstly, knowledge transfers within members of a project. Secondly, knowledge transfers from one project to another being conducted at the same time. Thirdly, knowledge generated from past projects should be made available to future endeavors. Finally, sensemaking is the act of comprehending knowledge retrieved from the system. This involves recognizing the contextual factors surrounding knowledge and trusting the source of insights.

4.6 Project management outcomes

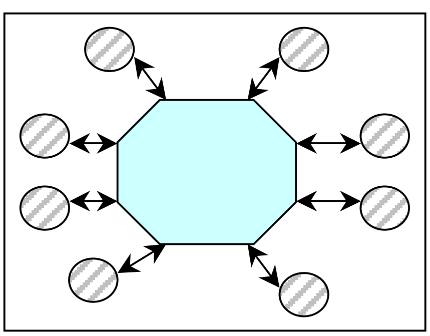
Aggregation, transfer, and sensemaking of knowledge in, from, and about projects has direct bearing on project management. If knowledge in projects is exchanged, transferred, and utilized effectively and efficiently we can expect high project performance. Knowledge about projects aids in evaluation and allocation of resources efficiently, which can increase an organization's performance if carried out efficiently. Proper coordination mechanisms would enable efficient transfer of knowledge, which would increase the satisfaction among decision makers due to symmetry of information available. Aggregating knowledge from projects is directly tied to organizational learning as insights can be used in future endeavors.

We now look at two common designs of knowledge management system in relation to the above mentioned factors, and their associated effects on knowledge management.

5. Centralized vs. Decentralized Approaches

Many knowledge management initiatives rely on information technology as an important enabler. The use of information technology for knowledge management has been approached from two angles, codification and personalization. As mentioned earlier, we can draw parallels between the codification and personalization strategies with client-server and peer-to-peer computing paradigms. We now examine these two approaches.

Figure 2: Client-Server Approach



5.1 Centralized Approach

As depicted in figure 2, in a client server approach to knowledge management, we have a central repository for organizational knowledge. Individual agents in the organization access the knowledge base for retrieval and update purposes. In client-server computing, a clear distinction is made between being a client or a server. In this approach, individual nodes in the network request services from a server. Clients cannot own the resources on the server. They are allowed to use it, but the control of the resources is with the server and not with the client.

Analyzing the effect of a centralized system on the aggregation, transfer, coordination, and sensemaking of knowledge in, about and from projects, we infer the following:

Capturing *knowledge in projects* into the central repository is of pivotal significance. As reported by many studies, members of the organization have fears that sharing their knowledge with the community at large make them less valuable to the organization. As such, the idea of contributing to a central repository does not jive well [6]. Moreover, there are inherent delays between the moment when the knowledge is created in the minds of the individuals and when it is posted on to the repository. The reason for this delay is that individuals are so overly conscious of the value of their knowledge and perception of their peers that they sometimes leave knowledge unposted. This defeats the concept of real time availability of knowledge, as insights not captured immediately are lost. Furthermore, many people under value themselves and are afraid to showcase their knowledge to a wider audience. They may have insightful knowledge that is not validated. Hence, they are afraid to post potentially invaluable insights on a central repository due to the fear that it may be incorrect. Yet another barrier for efficient knowledge capture is that the centralized approach detaches the contributor from his/her knowledge. Once stored centrally, knowledge seekers seldom are inclined to contact the knowledge provider unless clarity is needed, and unless one queries the source of knowledge, the provider goes unnoticed. Once posted centrally, the author loses control over who views the knowledge that he/she contributed. In centralized systems the extent of customization is limited; hence all project teams are bound by standards for inputting knowledge in repositories. These issues affect aggregation of knowledge in projects.

Knowledge contained in the central repository is structured on various dimensions such as by teams, products, divisions, etc, enabling faster access times to required elements. Moreover, it facilitates the use of filtering and categorizing mechanisms for sifting. However, the nature of centrality calls for use of global filtering and categorizing schemas, which are not optimal in all cases. Setting global thresholds for relevance, accuracy, and other attributes for knowledge may lead to loss of knowledge, as insights considered important for one project may be lost in the filtering process. The knowledge being captured is also forced into categories, resulting in better access times. One must also be aware that there are significant costs involved in finding the right category to store one's insights, which could act as a deterrent. When the knowledge content is highly structured, a great deal of effort is required up front to ensure the appropriate structuring. Employees would be required to append appropriate key words and metadata to their knowledge prior to posting the knowledge. They are in essence creating answers for questions that have not yet been posed and are implicitly required to address the issue of what queries should this knowledge be a result of. Hence motivation is a problematic issue since extra effort and time is required for structuring contributions.

Owing to its centralized nature, members in the organization know where the knowledge repository resides and have the prerequisite tools and knowledge to access it, thus making for ease of use. As the repository is built with the premise that all members of the organization need to access the knowledge base, care is taken to provide a shared context. This entails knowledge contributors taking extra effort to make sure their thoughts and insights can be understood by their peers once entered into the knowledge repository. Developing a ranking mechanism to indicate the relevance of results becomes easier owing to the structured categorization of knowledge and the shared context. Such a shared context facilitates sensemaking. In a centralized approach, identification of the knowledge source is easier and facilitates better coordination. Hence transfer of knowledge in projects from the provider to the consumer is improved by the centralized approach.

A problem that arises in the centralized approach is that much of the knowledge in projects would be local to one site or to one project and hence storing of such information on a central repository would be meaningless and irrelevant to non-members of the project. Since knowledge in projects is updated often, in many cases daily, to record details such as project schedules, milestones, minutes of meetings, training manuals, etc, inputting all the information to the central repository becomes cumbersome and costly. Hence, adopting a centralized approach to knowledge management would lead to overfilling of the knowledge repository increasing irrelevant search results.

Knowledge about projects is structured knowledge that can be captured and filtered efficiently in the centralized approach. Requirements for knowledge about projects do not change on a frequent basis. As such, having structured approaches for retrieval is facilitated via a centralized approach. However, only a small percentage of the organization uses knowledge about projects for budget preparation, staff allocation, and for other control purposes. Hence, storing such knowledge in a central repository is of minimal value to the remaining employees, who form the majority of the organization.

Inputting the *knowledge from projects* into a central source becomes time consuming, as a central repository would not render itself to easy customization. Individuals need to find appropriate categories and sections to contribute knowledge. One may not be able to contribute knowledge if a category does not exist, or one may need to exact effort trying to find the right category, and may view the costs of such efforts greater than the benefits derived from knowledge contribution. This would hamper the motivation of the employees for sharing their knowledge from projects due to the high costs involved in inputting insights into the system.

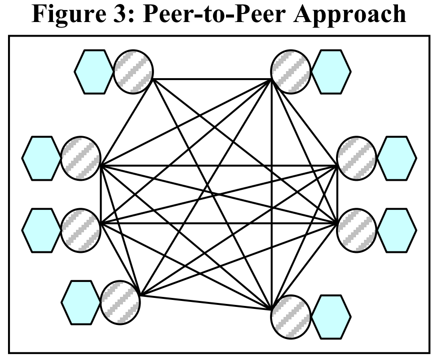
In non-collocated projects, the number of stakeholders involved may be one, or multiple. Coordination of the different people involved in a project becomes more important when multiple stakeholders are involved. In projects involving a single stakeholder, ownership of the central repository is not a problem. One common issue when having multiple stakeholders and utilizing a centralized approach to capture the three different types of knowledge is the question of ownership of the knowledge captured. The decision on the location of the central repository could be a potential problem. Hence, having one central repository to store all its knowledge poses a problem.

5.2 Decentralized Approach

As depicted in figure 3, in a peer-to-peer networking model, each node acts as both a client and a server; as such they can request services from other nodes in the network and also serve contents or services. Peer-to-peer computing is enhanced by the current availability of mobile devices using which ad hoc meetings can be set up easily to clarify assumptions, take decisions, justify decisions etc.

Analyzing the effect of such a decentralized system on the aggregation, transfer, coordination, and sensemaking of knowledge in, about and from projects, we infer the following:

Figure 3: Peer-to-Peer Approach



In the decentralized approach, each member of the organization retains his/her knowledge and has explicit control over it. Members are connected to their peers in the organization, and can choose what knowledge to share. Members can also view knowledge held by their peers and request it. This decentralized nature of the system addresses the main limitations of the client-server approach. Since individuals have control over their own knowledge repositories, they are less likely to view sharing of knowledge as a threat to their value. Moreover, one needs to identify clearly the source, prior to knowledge receipt; it serves as an enabler for providing incentives. One's knowledge can be valued by peers and appropriate incentive schemes can be devised to reward knowledge providers.

The overzealous concern that one's knowledge needs to be put in good form prior to sending it to the repository is minimized, as compared to the centralized approach. Hence there is less of a delay in knowledge development, as individuals are more likely to store rough notes, working documents, etc of insights on their local repositories than on the main server. This makes for more real-time knowledge. Since *knowledge in projects* varies depending on different projects, knowledge capture is facilitated as knowledge providers can develop their own customization when contributing knowledge such as creating a new category to input their knowledge. This accounts for flexibility when contributing knowledge. P2P schemes have the advantage of lack of dependence on the central repository. Valuable knowledge will have a tendency to propagate itself through the network. For instance, if Agent 1 downloads valuable knowledge from Agent 99 in the network, we now have two sources for the same insight, Agent1 and Agent 99. Hence, when Agent 39 searches for the same piece of knowledge he/she has two options from where to retrieve it. Over time, valuable knowledge will propagate throughout the network thus enabling load balancing and lack of dependency [16]. One potential problem when opting for a P2P approach is that when a person adds to the knowledge that he/she downloads, there are two versions of the knowledge, one that has been updated and one that hasn't been updated. This may lead to version control problems in the network.

Each agent may choose to invoke his/her unique coding and categorization scheme for knowledge, making availability of shared context impossible. Such a system will become ill structured over time, resulting in cumbersome search and seek times and irrelevant knowledge search results. This will hamper sense making. The ease of effort through which knowledge can be made available to the network can act as a double edged sword. While it ensures more real-time capture and dissemination of knowledge, it also leads to quality and validation issues. Premature ideas and thoughts may not be validated and hence be low contributors to one's quest for knowledge. This would also result in over filling of the knowledge repository. Locating the source of knowledge will pose a serious problem in this environment due to its distributed nature. For example, consider a case wherein a current project is similar to one conducted in the past. Current project members need to access knowledge from the past project, in order to execute tasks. Though they may know that such knowledge exists, finding where it exists would be a strenuous task. The problem is magnified when we consider multiple stakeholders and sites. Hence diversity of content can become overwhelming as it may become difficult to obtain the appropriate sources of knowledge due to information overload and lack of shared vocabulary.

When capturing *knowledge about projects*, in addition to difficulties in filtering and categorization, coordination becomes a problem. This is due to the necessity to integrate knowledge stored in different nodes to get a holistic perspective. This, in turn, affects sense making. *Knowledge from projects* may involve insights that cannot be categorized under one specific category. In such cases, each team can choose its own categorization scheme for knowledge. Such knowledge can be captured in P2P systems as they foster dialogue between the various agents of the team and develops a spirit of community, as each agent must interact with his/her peers to gain knowledge. A P2P knowledge management system fosters socialization and externalization by mandating interaction between peers.

When multiple stakeholders are involved, the question of ownership of knowledge arises. In the absence of a centralized repository, who owns the knowledge generated in, about, and from projects?

6. Hybrid Approach

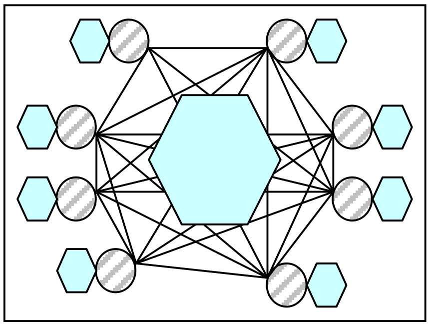
To gain benefits of the above two approaches and overcome the limitations we propose a hybrid model as depicted in figure 4.

At the core of the system we posit the need to have a central repository that holds popular knowledge (knowledge about and from projects) and serve as an index to knowledge (knowledge in projects) available by peers. By storing knowledge about and from projects in a central repository, we ensure the following:

1. Maintenance of a shared context. Exploiting such knowledge is easy.
2. Ease of access as knowledge about projects is well structured and stored in a central repository.
3. Ease in transfer of knowledge from projects through out an organization.
4. Validity of knowledge from projects as only validated knowledge will make its way to the central storage. Hence, members whose knowledge is stored centrally can also be rewarded for its high value.

5. Easy identification of the source of knowledge about and from projects.

Figure 4: Hybrid Approach



The second component of the central repository will hold an index of knowledge in projects available on the network. Having a centrally located index will enable fostering of an efficient coordination mechanism for knowledge in projects. It can serve as the knowledge dictionary to integrate individual knowledge. Common terms and categories can be assigned along with facilities to serve as an organizational thesaurus.

We assert that knowledge in projects must be exchanged via P2P approaches. Knowledge capture becomes efficient and effective as each project team can setup their own protocols; build their categories, setup filtering mechanisms, etc. Since each project is unique and each team is different, allowing for setup of flexible knowledge creation and exchange protocols is pivotal. In the context of non-collocated projects, each project team or site can have their own customization, which would be followed by other members of the team distributed over multiple sites. These local parameters will not affect the global knowledge of the organization. Moreover, it will promote efficient sharing of knowledge between members of the team, as the fear of making one's insights available throughout the organization is absent. And the knowledge captured from the different teams is indexed in the central repository. This would prevent overfilling of the central repository, while not losing out on relevant knowledge. Hence if an employee is interested in a particular knowledge, he or she can refer to that index and obtain the source of the actual knowledge. On locating the source, the employee can then request that knowledge directly from the source. Such a system gives control to the distributed server to service such a request or not.

In the realm of filtering and categorization, each distributed node develops its own local filters to filter and its own categories to categorize the captured knowledge. Local filters would prevent the loss of knowledge considered relevant to that project or site or stake holder. Hence, when a relevant index is located on the central server and the individual connects to the local server, search and access times are considerably reduced and relevance of search results is facilitated. Further, purging of the distributed knowledge repository becomes simple as each project team is given the right to purge their local repositories periodically. And each team can decide which knowledge is relevant and useful and which are not. This would prevent irrelevant and outdated search results and enable better access times. This would also help in circumventing the version control problem by updating their repository constantly, while purging out the old and irrelevant knowledge.

In non-collocated environments, such a hybrid model has greater utility. Consider two cases, involving single and multiple stakeholders. When a single stakeholder is involved and multiple projects are spread across multiple locations, each project develops its own structure of the repository and follows that design in all its locations. On the other hand, when multiple stakeholders are involved, the stakeholders of a project decide the kind of knowledge to be stored, its structure etc. Following this discussion, distributed repositories are set up for managing the knowledge.

One point worth noting is that the distributed knowledge repositories are developed on a per project basis across single or multiple stakeholders to store knowledge. Developing a common design or structure for such a repository may involve multitude deliberations and negotiations when multiple stakeholders are involved, but arriving on a consensus is much easier than when following a centralized system as the benefits outweigh the costs involved in such a lengthy process. Though this process may be time consuming initially, such an approach ensures better coordination among the participating members and efficient use of the knowledge being captured.

7. Conclusion

In the paper, we have addressed the issues of knowledge management systems in non-collocated environments. We have specifically analyzed two common approaches to knowledge management and made a case for a hybrid model. Our insights have implications for practitioners, as one can consciously choose the right scheme for managing knowledge in projects, and also knowledge about and from projects. For researchers, we have laid the foundation for inquiry into some key issues in distributed knowledge management. Questions of interest include empirically testing the theoretical framework using the different systems approaches. The role of context can be studied examined both from the global and local perspectives. Using theories such as Task-technology fit and transaction cost economics one can see if the above models can be shown to work well with a certain class of organizations or projects. The unique contribution of this paper is the analysis of how various system architectures effect knowledge exchange in, from, and about projects

9. References

[1]Brown, J.S. & Duguid, P. (1991) "Organizational learning and communities of practice: Towards an unified view of working learning and innovation," *Organization Science*, 2(1), 40-57.

[2]Chattaraj, A., Desouza, K.C. & Kraft, G. (2002). "Knowledge management in organizations: Suggested research insights from systems, information, and bounded rationality theories," unpublished working paper, Illinois Institute of Technology, Chicago, IL.

[3]Damm, D. & Schindler, M. (2002). "Security issues of a knowledge medium for distributed project work," *International Journal of Project Management*, 20, 37-47.

[4]Desouza, K.C. (2001). "Intelligent agents for competitive intelligence: Survey of applications," *Competitive Intelligence Review*, 12 (4), 57-63.

[5]Desouza, K.C. (2002a). "Facilitating tacit knowledge exchange: A humanistic and entertainment approach," *Communications of the ACM*, forthcoming.

[6]Desouza, K.C. (2002b). "Barriers to effective use of knowledge management systems in software engineering," *Communications of the ACM*, forthcoming.

[7]Desouza, K.C. (2002c). *Knowledge management with artificial intelligence*, Westport, CT, London: Quorum Books.

[8]Dougherty, D. (1992). "Interpretative barriers to successful product innovation in large firms," *Organization Science*, 3(2), 179-202.

[9]Evaristo, J.R. (2001) "Nonconsensual negotiation in distributed collaboration," *Communications of the ACM*, 44(12), 89.

[10]Evaristo, J. R. & Fenema, P. (1999). "A typology of project management: Emergence and evolution of new forms," *International Journal of Project Management*, 17(5), 275-281.

[11]Grant, R.M. (1996). "Toward a knowledge based theory of the firm," *Strategic Management Journal*, 17, 109-122.

[12]Hansen, M. T., Nohira, N. & Tierney, T. (1999). "What's your strategy for managing knowledge?" *Harvard Business Review*, 77(2), 106-116.

[13]Hinds, P. & Kiesler, S. (1995). "Communication across boundaries: Work, structure, and use of communication technologies in a large organization," *Organization Science*, 6(4), 373-393.

[14]Katzy, B., Evaristo, J.R. & Zigurs, I. (2000). "Knowledge management in virtual projects: A research agenda," In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Maui, Hawaii.

[15]Nonaka, I. & Takeuchi, H. (1995). *The knowledge-creating company*, New York, Oxford University Press.

[16]Parameswaran, M., Susarla, A. & Winston, A.B. (2001) "P2P networking: An information-sharing alternative," *IEEE Computer*, 34 (7), pp.31 – 38.

[17]Ramaprasad, A. & Rai, A. (1996). "Envisioning management of information," *Omega*, 24(2), 179-193.

[18]Ruggles, R. (1998). "The state of the notion: Knowledge management in practice," *California Management Review*, 40 (3), 1998, 80-89.

[19]Turner, J.R. (1993). *The handbook of project-based management*, Maidenhead: McGraw Hill.

[20]White, D. & Fortune, D. (2002). "Current practice in project management - An empirical study," *International Journal of Project Management*, 20, 1-11.