

Understanding and Improving Collective Attention Economy for Expertise Sharing

Yunwen Ye^{1,3}, Kumiyo Nakakoji^{1,2}, Yasuhiro Yamamoto²

¹ SRA Key Technology Laboratory, Inc., 3-12 Yotsuya, Shinjuku, Tokyo 160-0004, Japan

² RCAST, University of Tokyo, 4-6-1 Komaba, Meguro, Tokyo, 153-8904, Japan

³ L3D Center, University of Colorado, Boulder, CO80309-0430, USA

yunwen@colorado.edu; kumiyo@kid.rcast.u-tokyo.ac.jp; yxy@kid.rcast.u-tokyo.ac.jp

Abstract. The importance and benefits of expertise sharing for organizations in knowledge economy are well recognized. However, the potential cost of expertise sharing is less well understood. This paper proposes a conceptual framework called collective attention economy to identify the costs associated with expertise sharing and provide the basis for analyzing and understanding the cost-benefit structure of different communication mechanisms. To demonstrate the analytical power of the conceptual framework, the paper describes a new communication mechanism—Dynamic Mailing List (DML)—that is developed by adjusting certain cost factors.

Keywords: collective attention economy, expertise sharing, socially aware

1 Introduction

Despite the advance of information systems that makes it easier to store and access knowledge, knowledge workers who are faced with complex knowledge-intensive work still routinely rely on their peers for knowledge and expertise [1]. Knowledge held by its members constitutes one of the most important assets of an organization. To fully utilize such valuable assets, one of the key challenges in the design of information infrastructure for organizations is to facilitate the easy transfer, sharing and integration of knowledge held by members with computer-mediated communication mechanisms that connect an expertise seeker, who is looking for specific knowledge to solve his or her own problem, with an expertise provider, who holds the sought after knowledge. Much research on expertise sharing has been conducted on providing support for expertise seekers, through the design of different mechanisms of finding who have the needed knowledge to become expertise providers and who are available to help [3].

Despite its great importance and benefits, expertise sharing, however, comes with a cost. When an expertise seeker posts a question, a *seeking cost* occurs because the expertise seeker has to spend some time to formulate the question and to decide whom to

ask. When an expertise provider offers an answer, an *answering cost* occurs because the expertise provider has to pay attention to read the question and formulate an answer. The real cost, however, could be far greater than the sum of the *seeking cost* and the *answering cost*, depending on the communication mechanisms used. For example, if a community-wide communication mechanism such as mailing list is used to post the question, all members of the community who pay any form of attention—from receiving the question, scanning the subject, skimming the contents, to answering the request—have paid a cost of attention, which is a scarce resource. Collectively, the cost of expertise sharing could be very high, and even outweighs its benefits and lower the group productivity [17].

To understand the cost of expertise sharing better, we introduce the notion of *collective attention* to denote all the attention that is consumed by all parties involved in a transaction of expertise sharing; and present a conceptual framework called the *economy of collective attention* to analyze the cost-benefit structure of communication mechanisms used for expertise sharing. The framework guides us to articulate design requirements for information systems in support of *situated expertise sharing* to improve the utilization of collective attention. We use the term *situated expertise sharing* to refer to a particular kind of expertise sharing in which a knowledge worker asks peers questions for the purpose of obtaining the specific knowledge that is needed in his or her own work. A new communication mechanism called Dynamic Mailing List (DML) is introduced and its implementation in a system is briefly described.

2 Problem Context: Situated Expertise Sharing

Expertise sharing takes place in many different situations and for different purposes. Despite many shared aspects, the practice of sharing expertise faces different challenges when the types of knowledge and the external constraints vary. Our research focuses on expertise sharing situations in which a group of knowledge workers engage, under strong pressures of productivity and quality, in the collaborative construction of a common knowledge artifact that is decomposed into parts that have complicated inter-dependency. The common artifact is made possible through the integration of distributed work and knowledge of each knowledge worker. Each worker is responsible for constructing some parts by bringing their unique set of knowledge. Due to decomposition, each knowledge worker only has partial knowledge of the artifact and of the process; and due to the inter-dependency, each knowledge worker often needs to seek knowledge from peer workers of the same group to carry out his or her work efficiently and effectively.

This kind of *situated expertise sharing* is not for the general purpose of learning or creating awareness in which knowledge is not immediately coupled with the task at hand. Rather, it is a clearly purposed act that serves the goal of the accomplishment of an individual worker's current task, and it arises on an as-needed basis and requires quick resolution. Software development is one typical example. Programming requires undivided attention, and in general programmers prefer to work in solitary with long

periods of uninterrupted time during which they can concentrate. It is this kind of solitary work that gets the code written. However, due to the interdependency of their work, programmers also have to engage in situated expertise sharing with peers to seek knowledge necessary to accomplish their individual task. A study has shown that this type of ad hoc and situated expertise sharing takes up to 41% of a programmer's time [19].

The above settings entail the following constraining factors that have to be balanced by information systems that support situated expertise sharing:

- (1) *The collective attention has a limited capacity due to the fixed size of the project group.* This is different from volunteer-based community projects such as Wikipedia where the number of contributors can be increased through strategies of turning passive users into active contributors.
- (2) *Situated expertise sharing is not a one-time affair; it has to be sustainable* because its continuous enactment is required throughout the lifecycle of the project. A member's engagement in one sharing act should not result in his or her reluctance to participate in further sharing acts down the road. No absolute experts exist. Depending on the context, a knowledge worker often assumes the role of seeker or provider of expertise at different times.
- (3) *The costs and benefits of situated expertise sharing have to be considered together with the group productivity.* On the one hand, if a knowledge worker is unable to obtain the knowledge in the head of peers timely, he or she cannot carry out his or her task effectively, and thus lowers his or her own productivity, which in turn lowers the productivity of the project group. On the other hand, if a knowledge worker is frequently interrupted for offering help, the expert's productivity is significantly reduced, resulting in lower group productivity. It has been observed that such costs could even outweigh its benefits and lower the group productivity [17].

3 Collective Attention Economy

Attention is directed, involved awareness, and is the intellectual energy invested toward some purpose. Attention is an intrinsically scarce resource because everyone has only a certain stock of supply. In this Internet era full of overwhelming information, many of us feel short of enough attention. Goldhaber and others have eloquently argued that we are entering a world where our lives are guided more by the laws of the economics of attention because attention is quickly becoming the scarcest resource in our society [9].

Attention economy is concerned with the use or the patterns of allocation of attention for the best possible benefits. Collective attention economy is concerned with the effective use of the sum of attentions of members in a group. To improve collective attention economy within a group is to improve the patterns of attention allocation of its members for the purpose of achieving better expertise sharing results and increase the collective attention capacity of the group.

3.1 The Cost of Collective Attention in Situated Expertise Sharing

In an act of situated expertise sharing, both the *asker* (expertise seeker) and the *recipients* (those who receive the question asked by the asker) consume attention. Those recipients who provide answers to the question are *helpers*. Those recipients who do not provide answers are *onlookers*. Helpers consume more attention than onlookers.

An asker needs to find out who has the expertise. Previous research has shown that such awareness of who knows what takes extensive time to develop, and its utilization consumes intensive attention [13]. We denote the attention cost as C_{Find} .

The question needs to be formulated and articulated, and we denote this attention cost as C_{Ask} . Rhetorical strategies, linguistic complexity and word choice of the question all influence the likelihood of receiving replies to a question [4, 5]. The asker also needs to make a decision based on social cues whether the potential experts could be interrupted, and to determine opportune times to interrupt [6].

When the recipients receive the question asked by an asker, all of them are interrupted and distracted from their current work. The cost of attention, denoted as $C_{Interrupt}$, includes not only the attention spent on attending to the interrupting event but the disruption of flow and the accompanied work resumption efforts [10, 23].

Recipients need to make a conscientious decision to respond to it or not. A number of factors affect this decision-making process: whether they have sufficient expertise or interest on the topic [24]; how they value their contributions by answering the question [16]; how many efforts does it take to post a reply [11]; how they perceive their relationship with the asker [5]. To make this decision, they at least need to skim the question by finding out who is the asker and what is the topic [10]. We denote this attention cost as C_{Skim} , and this cost applies to all recipients.

If a recipient decides to respond to the help request, he or she needs to spend time and attention in thinking and composing the response. The cost of attention for answering the question is denoted as C_{Answer} , and this cost is incurred only on helpers.

Upon receiving an answer, the asker needs to evaluate its quality and interpret its meaning in terms of his or her task. Not all responses are of equal value, and some of them may not be very helpful. We denote this cost of attention as $C_{Evaluate}$.

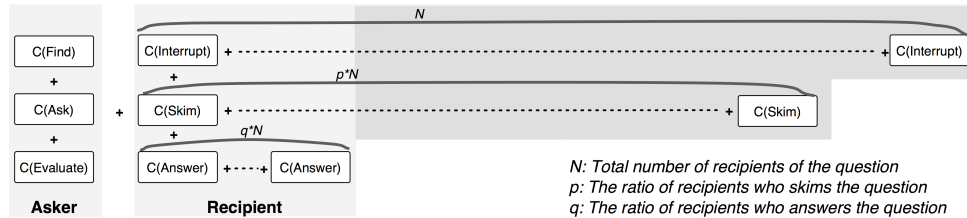


Fig. 1. Cost of Collective Attention (CoCA) in Situated Expertise Sharing

Fig. 1 shows the total *cost of collective attention* (CoCA for short) for an act of situated expertise sharing. Suppose the question posted by the asker is sent to N recipients, then all

the recipients will have to shoulder a cost of $C_{Interrupt}$. Among the N recipients, some of them (we denote the ratio as p) will read the question, and some of them (we denote the ratio as q) will answer the question.

3.2 Approaches to Improving Collective Attention Economy

Two major variables affect the effectiveness of the total collective attention consumed for situated expertise sharing. The first variable is the total number of recipients (N). If only the experts who can provide helpful answers receive the question, then the attention is well spent. The second variable is the success rate (denoted as r) of expertise-seeking attempts because only successful expertise sharing acts return benefits. The two variables are not independent: r tends to increase together with N because when N increases, the possibility of someone who is able and willing to answer increases. However, when N increases, the $CoCA$ for each act of expertise sharing also increases.

Improving collective attention economy can be approached from two directions. First, one can try to raise the success rate r by increasing the ratio of posted questions that receive timely and high quality replies over the total number of questions that are asked. Second, one can try to reduce $CoCA$ for each act of expertise sharing. The most significant waste of attention is that consumed by onlookers: $(1-q)*N*C_{Interrupt}+(p-q)*N*C_{Skim}$: the attention spent by those who are interrupted or read the question but are not interested in providing answers (the darker area in Fig. 1).

One might argue that these attentions are not wasted because by reading only albeit not actively participating, the onlookers also might learn something useful for the future. Moreover, such seemingly wasted communication often contributes to increasing shared context and awareness. However, shared awareness comes at the cost of collective attention cost. In general, knowledge workers “are ‘not interested’ in the enormous contingencies and infinitely faceted practices of colleagues unless they may impact our own work ” and workers “routinely expect not to be exposed to the myriad detailed activities” of others [21].

Our goal is to pursue the right balance among the degrees of awareness, connectedness, and the needs for solitary concentration craved by knowledge workers. To reduce $CoCA$, our approach is to find a way to reduce N without reducing r . At the same time, we try to complement the plausible side effect of decreasing shared context and awareness through other means, such as the accumulation of discussion archives which can be used as a means to provide contexts and as learning materials by those not directly involved.

4 The Dynamic Mailing List Approach

This section describes the Dynamic Mailing List (DML) mechanism that we have designed to improve the collective attention economy for situated expertise sharing.

4.1 Basic Strategies

Our approach is based on the following strategies:

- Keep C_{Find} to be zero; namely, expertise seekers can ask questions without knowing or thinking of to whom it should be sent.
- Reduce the number of those who receive the question with minimal effects on r . We use heuristic strategies to route questions to the people who are most likely and willingly to answer the question, and reduce the wasted attention of onlookers.
- Raise the ratio r by reducing the number of unsuccessful expertise sharing acts through the improvement of the quality of each act of expertise sharing.

Due to the wide diversity and fine specialization of knowledge, not all members in a group have expertise or are interested in all topics discussed in all acts of situated expertise sharing. The attention ($C_{Interrupt} + C_{Skim}$) consumed by those members who receive the question but have no expertise on the specific topic of the question is underutilized. Even if an answer is offered, the answer is often not helpful due to the lack of matching expertise, and it also increases the cost of attention ($C_{Evaluate}$) consumed by the asker in evaluating the answers because he or she has to sift through all answers to find the useful ones. This leads to the first design principle for the DML mechanism.

Principle 1: *Members who do not have matching expertise shall not receive the question.*

Having expertise is only a necessary not a sufficient condition for a peer to share his or her expertise: the peer has to be willing to share the sought after expertise. Some knowledge workers might not want to provide expertise on certain topics for various reasons. Some members might get bored by answering repeatedly questions that they deem too simple to worth their time and expertise; and some might want to guard their certain expertise to retain their “market value” in the organization [18]. When workers are forced into sharing expertise they are not willing to share, they often use “verbal and intellectual skills as a defense to keep a person with a problem from consuming too much of their time,” and their answers are often “impressive-sounding” but not helpful [5]. As a result, the attention consumed by both the expertise seeker and the unwilling expertise provider is wasted. These empirical observations lead to the second design principle.

Principle 2: *Members who are not willing to share the matching expertise shall not receive the question.*

Another factor that affects a recipient’s willingness of sharing expertise is his or her perceived social relationships with the asker and with the group at large. Favorable interpersonal relationship facilitates expertise sharing due to preexisting trust and mutual understanding [2]. When an expert chooses to pay attention to an asker, the social relationship between the two people gets changed. A recipient decides to help the asker based on his or her perceived existing relationship or desired future relationship with the asker. Given the varieties of human relationships in a group, it is natural that a member may want to be related only with a certain subset of the whole group. Because an arduous relationship between an expertise seeker and an expertise provider often leads to the

failure of expertise sharing [5], a recipient who does not like to work with the asker is not likely to provide help. People have very nuanced preferences concerning how and with whom they like to share expertise and like to maintain control of their social interaction [1]. Hence we have the following design principle.

Principle 3: *Members shall be able to decide with whom to share expertise.*

4.2 Creating Dynamic Mailing Lists for Situated Expertise Sharing

A dynamic mailing list (DML) differs from traditional mailing list in that a new mailing list is created every time when an asker posts a question, with the recipients decided dynamically based on the three principles identified in section 4.1. The bottom half of **Fig. 2** shows the general architecture of a server that supports the DML approach. The top half shows an illustrative use scenario. Suppose *Harry* has a question about a Java API method named *exec*. He first searches the *Discussion Archives* that store previous discussion emails exchanged through the DML server. If Harry does not find answers in archives, he can post a question, and the DML server uses two steps to decide the recipients of the question: *expert identification* and *expert selection*.

Based on the *Human-Topic Relationship* database, which stores the *expertise profiles* for members, the expert identification step creates a list of *candidate experts* who have expertise on the topic of the question (*exec*). The expertise profile of a member represents what topics on which the member has expertise. A member can manually specify his or her expertise, but this is usually too costly. In practice, this expertise profile should be automatically generated using data mining techniques by analyzing preexisting documents, artifacts and expertise-sharing activities. Following Principle 2, this expertise profile should also include topics on which a member is not willing to share his or her expertise.

From the list of *candidate experts*, the expert selection step chooses, based on the *Human-Human Relationship* database that represents the social relationships among members, those who have the highest possibility of helping the asker (*Harry*). A member's social relationships with other members and the group are represented by his or her *social profile* that consists of the following four kinds of basic relationships:

- *help* $\langle A, B \rangle$: This represents how many times that *A* has provided answers to questions posted by *B*.
- *include* $\langle A, B \rangle$: *A* states that *A* is willing to help *B* if he or she has any level of expertise on a topic that *B* asks.
- *exclude* $\langle A, B \rangle$: *A* states that *A* is not willing to share expertise with *B* on whatever topics.
- *email* $\langle A, B \rangle$: This represents the number of regular emails that *A* has sent to *B* outside of the DML server. This is extracted by analyzing the mailbox of each member, and represents the existing social relationships that *A* and *B* had before they started using the DML approach.

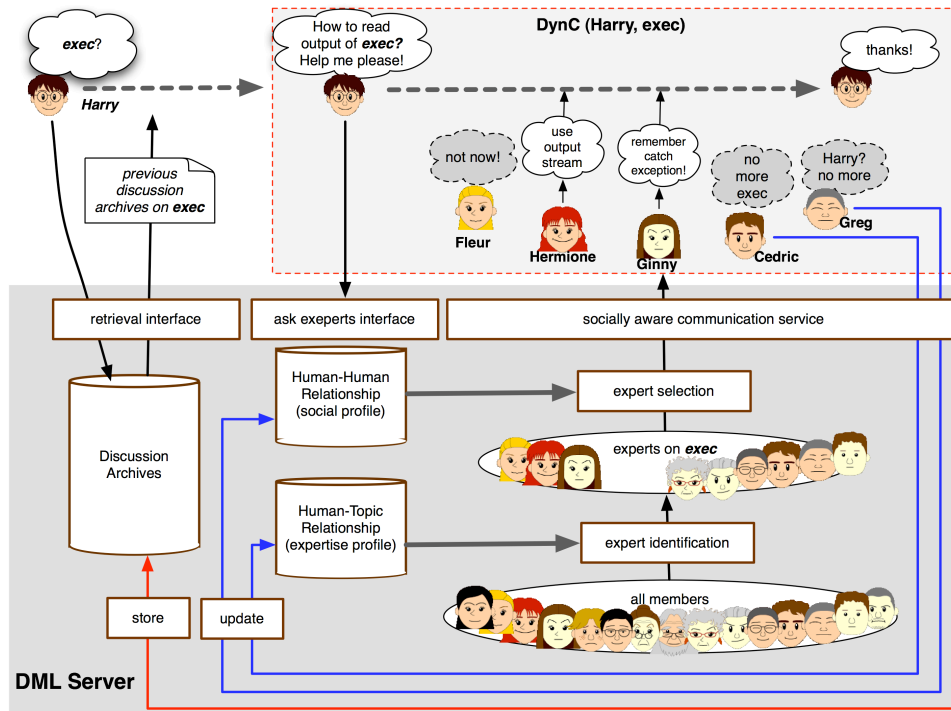


Fig. 2: The general architecture of Dynamic Mailing List and an illustrative scenario

With the relationships defined above, the expert selection step in DML uses the following process to choose from the list of *candidate experts* a predefined number of members who are most likely to answer *Harry*'s question on *exec*. The selection process consists of 5 passes, and stops processing the next pass when the predefined number of experts has been selected.

Pass 1: Based on the preference of unfavorable relation. For each person X in the list of *candidate experts*, if $exclude\langle X, Harry \rangle$ exists, X is removed from the list of *candidate experts* because X has indicated no intent to help *Harry* on whatever topics.

Pass 2: Based on the preference of friendly relation. For each X in the remaining list of *candidate experts*, if $include\langle X, Harry \rangle$ exists, X is selected because X declares his or her willingness to help *Harry*.

Pass 3: Based on the rule of direct reciprocity. For each X in the remaining list of *candidate experts*, if *Harry* has helped X more times than X has helped *Harry* (i.e. $help\langle Harry, X \rangle \geq help\langle X, Harry \rangle$), X is selected because social norms in general requires that X reciprocate the favor that he or she has received from *Harry* in the past.

Pass 4: Based on the rule of generalized reciprocity. From the remaining list of *candidate experts*, this pass chooses those who have been helped more in the whole group

than they have helped others, regardless their direct relations with *Harry*. Although they may not bear direct social relations with *Harry*, they should offer help due to the principle of generalized reciprocity in the whole group.

Pass 5: Based on outside communication. If no sufficient number of experts has been selected till this pass, the DML mechanism resorts to preexisting social relationships that members have before they started using DML. From the remaining list of *candidate experts*, this pass chooses the experts who have sent most emails to *Harry* based on $email\langle X, Harry \rangle$, because X must know *Harry* to a certain degree if he or she has sent emails to *Harry*, and might be willing to help *Harry*. Unlike the initial 4 passes that use the relationships resulted from the social interactions within the DML mechanism, this pass uses the social interaction history outside of the DML mechanism, and this pass is heavily used at the initial stage of the DML mechanism when there is no sufficient history of interactions within the DML mechanism.

The chosen experts become the participants of the dynamically created mailing list $DynC(Harry, exec)$, and will receive the question asked by *Harry* on *exec* through emails. If a recipient (X) replies to the question, his or her reply is sent to the same members of the $DynC(Harry, exec)$, and the DML server automatically increases $help(X, Harry)$ by 1 to update the social profile of X .

The dynamic mailing list is disposed either manually by the asker when he or she decides there is no more need for expertise sharing on the topic, or automatically by the DML server when there is no more email exchanges for a predefined period of time. The messages are stored in the Discussion Archives.

As a result of this two-step processing by the DML server, only those members who have both high level of expertise on the topic and high possibility of helping the asker are chosen to receive the question. By so doing, the success rate r of expertise exchange can be improved because the intensity of engagement by each expertise provider is directly related to his or her willingness to actively inquire into and understand the asker's problem and then shape their answer to the problem in generating help [5]. This would also translate into a lower cost of $C_{Evaluate}$ on the asker's side. Because the list of expert recipients is automatically generated, the C_{Find} remains to be minimal. $CoCA$ is reduced because those members who do not have the relevant expertise on the particular question or who are not likely to help the asker do not receive the question and do not need to consume their attention in receiving and reading the question. The potential learning benefits of the onlookers can be similarly achieved through the browsing and searching of the accumulated Discussion Archives.

4.3 Socially Aware Communication

When expertise providers specify their preferences on the kinds of expertise sharing activities they do not want to participate, they are saying "no" explicitly to their peers. In a community where social norm expects active collaboration, their decision of no participation may cause damages to relationships with other members and risk disruption

of group cohesion. Many people who feel guilty when they have to say no loudly and publicly might choose not to participate in the community all together or be forced to endure the waste of their attention repeatedly. Both deteriorate the economy of the collective attention, with the former decreasing the total available resources of expertise and the latter wasting the attention of onlookers.

Improving the economy of collective attention, therefore, needs to be complemented with socially aware communication mechanisms. Socially aware communication refers to the transmission of information or signals that does not violate social norms, and therefore is not punishable by the iron hand of social pressure that establishes group cohesion through enforcing the required individual behaviors [15]. In face-to-face communications, many non-linguistic social signals are used to encourage or discourage the further occupation of attention in a socially acceptable way, without causing unwanted damages to social relationships.

Although information and communication technology has advanced greatly in easing the transmission of non-linguistic social signals to approximating the effects of fact-to-face communication, it also offers opportunities of devising alternative ways of conducting socially aware communication other than those used by people in face-to-face communications. The DML approach adopts the following principle of *asymmetric disclosure of information* to achieve socially awareness: To make the decision of no participation socially acceptable, DML ensures that an expertise provider's refusal of further allocation of attention on unwanted expertise sharing be not known by other members, and therefore makes the refusal socially plausible.

First, all choices made by an expertise provider regarding his or her preference of participation in the kinds of expertise sharing must be strictly limited to the eyes of the expertise provider only. No other members should know the decision. This can be realized with login names and passwords. Second, when an asker posts a question through DML, the recipients of the question are not made public. Only the recipient knows that he or she gets the question. No other members, including the asker and other members who receive the same question, know who else receives it. On the other hand, if a recipient replies, his or her name is revealed to all participants of the DML, and is revealed to all other members in the Discussion Archives. This asymmetric disclosure of information removes the undesired social implications of no participation but highlights and encourages cooperative behaviors with explicit acknowledgement of participation. With the existence of such socially aware communication mechanisms, members can freely say no—actually they do not say it aloud, they only choose no secretly—if they are unable to help others at any given time.

Each question sent through a DML is associated with two links that allows its recipients to update his or her expertise profile and social profile. As shown in the scenario of Fig. 2, if a member finds that he receives a question on a topic that he or she does not want to answer any more, he or she can click one link to update his or her expertise profile so that he or she will no longer receive questions on the same topic in the future. If a member finds that he or she is receiving questions from a person he or she does not like to work with, he can click on the link to update his or her social profile to

add an *exclude* relationship so that he or she will not receive any question from the asker in the future. Both links reduces the burden of maintaining updated expertise profile and social profile. Updating profiles costs a little bit more attention than just ignoring the question, but this one-time extra attention cost will lead to the reduction of future attention cost of dealing with questions that the member does not want.

5 Implementing DML in the STeP_IN System

As a way to illustrate the DML mechanism, we have implemented it in the STeP_IN system (standing for Socio-Technical Platform for in situ Networking) [25, 26]. A huge reusable class library is one of the major benefits brought by Java, but it also poses great challenges for programmers to learn to use those library methods. Most programmers only know a portion of them, and the expertise of the library is asymmetrically distributed among programmers. This gives rise to many needs of situated expertise sharing that often take place in the middle of programming when a programmer needs to use an unknown API method. An effective way of learning is to ask those who are experts on the given method. The STeP_IN system implements the DML mechanism to help programmers to learn from their peers by asking questions about Java API methods. The system also provides other technical support, but this paper will focus on the issues related to the DML mechanism, for more details on other aspects of the system, please see [26].

draco's Expertise Profile

Profile Summary
Total methods used: 17
Methods declared as Expert: 0
Methods declared as No Knowledge: 0

Display methods...
All methods used

Options...
Scope: java.lang
Filter:

[Previous] 1 [Next]

Method	Author	Usage #	Declare
java.lang.Process java.io.OutputStream getOutputStream()	No	1	<input type="radio"/> Expert <input checked="" type="radio"/> Not Declared <input type="radio"/> No Knowledge
java.lang.Runtime java.lang.Process exec(java.lang.String)	No	1	<input checked="" type="radio"/> Expert <input type="radio"/> Not Declared <input type="radio"/> No Knowledge
java.lang.StringBuffer java.lang.StringBuffer append(java.lang.String)	No	1	<input type="radio"/> Expert <input type="radio"/> Not Declared <input checked="" type="radio"/> No Knowledge
java.lang.StringBuffer java.lang.String toString()	No	1	<input type="radio"/> Expert <input checked="" type="radio"/> Not Declared <input type="radio"/> No Knowledge

Done

draco's Social Profile

Your Social Profile is only known to you.

Profile Summary
You declared to participate in DynCs initiated by 1 members.
You declared not to participate in DynCs initiated by 1 members.
You have email exchanges with 5 members.
You have helped other members 1 times, and you have been helped 0 by other members.

Display members...
All members related

Options...
Scope:

[Previous] 1 [Next]

User Name	Participation in His/Her DynC	Participation in My DynC	Mail Exchange From	To	Future Participation in His/Her DynC
greg	0	0	0	2	<input type="radio"/> always <input checked="" type="radio"/> neutral <input type="radio"/> never
harry	0	0	0	1	<input type="radio"/> always <input type="radio"/> neutral <input checked="" type="radio"/> never
rita	0	0	0	3	<input type="radio"/> always <input type="radio"/> neutral <input checked="" type="radio"/> never
vincent	1	0	0	4	<input checked="" type="radio"/> always <input type="radio"/> neutral <input type="radio"/> never
yyc	0	0	0	2	<input type="radio"/> always <input type="radio"/> neutral <input checked="" type="radio"/> never

Fig. 3. Expertise and Social Profiles in STeP_IN

At the core of the DML mechanism is the creation and use of *expertise profiles* and *social profiles*. Because social profiles represents social relationships resulted from social interactions among members that are domain independent, the social profile in STeP_IN is defined in the same way as described in Section 4.2. It has four kinds of relationships: *help*, *include*, *exclude*, *email*. Expertise profiles are domain dependent. In STeP_IN, a programmer's expertise profile has two sets of Java API methods. The first set is *known*

methods, which include those methods of which the programmer has expertise. The second set is *uninterested methods*, which include those methods on which the programmer does not to share expertise with others.

To use STeP_IN, a user has to register first. Upon a user's registering to the system, an initial expertise profile for the user is automatically created by analyzing all the Java programs that he or she has written. The number of API method usage is extracted and stored in his or her initial expertise profile. A user can edit his or her expertise profile through the expertise profile management interface in STeP_IN (Fig. 3a). A user can select *Expert* in the *Declare* column to add the method to his or her *known methods* set no matter whether he or she has ever used it or not, or a user can select *No Knowledge* to add the method to the *uninterested methods* set in his or her expertise profile so that he or she will not receive questions about the method. As we have discussed in Section 4.3, such a selection can also be made once he or she receives a question on the method through DML emails, to ease the task of maintaining updated profiles.

At the registration time, an initial social profile is automatically created by analyzing the user's mailbox. It contains the number of emails that the user has received from other members. The other key element in the social profile is the *help* history the user has with other members. The number in the *Participation in His/Her DynC* column (Fig. 3b) indicates the number of help that the user has given to the member, whose name is shown at the first column; and the number in the *Participation in My DynC* column indicates the number of help the user has received from the same member. Through the profile management interface, the user can declare *include* and *exclude* relation with the member by choosing *always* or *never* in the *Future Participation in His/Her DynC*. Similarly to expertise profile, this personal preference can be made when he or she receives a question form the member through DML.

We will now continue to use the scenario in Fig. 2 to illustrate how a user interacts with the STeP_IN system for situated expertise sharing. When Harry posts a question on *exec* in STeP_IN, the system first creates a list of members whose *known methods* include *exec* and then removes from the list those whose *uninterested methods* contain *exec*. The resulted list is the list of *candidate experts* on *exec*.

From the list of *candidate experts*, STeP_IN chooses 5 members who have established social relationship with *Harry* based on each person's social profile. It first excludes those who declared *never* to participate in *Harry's* DynC. For example *Draco* would be removed because he have chosen *never* in his relationship with *Harry* (Fig. 3b) although he declared himself an expert on *exec* in his expertise profile (Fig. 3a). From the remaining list of *candidate experts*, the DML server in STeP_IN first chooses those whose social profile includes an *always* declaration regarding *Harry*; and then chooses according to the numbers appeared in the columns 2 and 3 in the social profile (Fig. 3b), i.e. those who owe *Harry* because they were helped more often by *Harry* than they helped *Harry*. If the system cannot choose 5 members from those who directly interacted with *Harry*, it chooses those who have received more help from the group. If the above process fails to reach the number 5, it uses the *email* relationship and chooses members who have sent most emails to *Harry*.

The question posted by *Harry* is then sent to the selected experts (*Fleur*, *Hermione*, *Ginny*, *Cedric* and *Greg*). However, *Harry* does not know who receives the question. Each recipient does not know who else receives the question. *Greg* finds the question is from *Harry* whom he is not fond of, so he clicks one of the embedded links that takes him to his social profile management interface and chooses *never* regarding *Harry*. *Cedric*, who finds he gets yet another question on *exec*, decides to change his expertise on *exec* to *No Knowledge* to avoid getting further questions on *exec*. *Hermione* and *Ginny* replied to the DML. *Fleur*, who is preoccupied with her own work, stays silent. Due to the asymmetric disclosure of information, all members know that *Hermione* and *Ginny* have helped *Harry*, but no one knows that *Fleur*, *Cedric* and *Greg* have received the question and *Cedric* and *Greg* have changed their preferences.

Harry is satisfied with the help he gets from *Hermione* and *Ginny* and goes to STeP_IN to evaluate the DynC as helpful. The DynC is then discontinued and the emails exchanged are archived and linked to the method *exec*.

An evaluation of the STeP_IN system [26] shows that the DML mechanism may miss some experts who are eager to help others regardless of their social obligations. This problem can be solved if the eager helpers set their participation preferences to *always* for all members so that they will be included in all expertise sharing acts concerning topics on which they have expertise. They can even choose *Expert* on all methods so that they will be included in all acts of expertise sharing. The DML mechanism will act like a traditional mailing list for members with the above generous settings, but other less eager helpers will still have their choices and options to control their allocation of attentions.

6 Concluding Remarks

Most of the research on expertise sharing has focused on helping users find the right expert [2, 7, 12, 13]. Such systems aim to reduce the cost incurred on the asker, mainly C_{Find} . Reder points out, however, that automated attempts to “pin people down” may not bring about better communication or enhanced productivity because successful expertise sharing requires intensive engagement of expertise providers [17]. Most existing research focuses only on the benefits an asker receives, and ignores the cost that helpers shoulder as well as the potential adverse impacts on group productivity. However many empirical studies have concluded that the cost of interruption [10, 14] and the overload of communication brought by ubiquitous connectivity [6] create the “dearth of attention”[22]. We do not have a systematic way to address the attention cost resulted from communication technologies.

In this paper we attempted to balance the needs of askers and the burdens of helpers in expertise sharing. The necessity to balance attention and communication are recognized in [8], which suggests two strategies to conserve attention resources in communication by providing information asynchronously and by reducing the frequency of interruption through the aggregation of information. These strategies can be subsumed in reducing the

cost of $C_{Interrupt}$ in the conceptual framework of the collective attention economy. However, as we can see from our analysis, this cost is only a portion of the cost of collective attention in collaboration.

We are fully aware that to model concepts as complicated and subjective as attention should not be taken lightly. The proposed notion of *CoCA* is not meant to compute the absolute value of attention cost. The main goal is to use this relatively simple framework to analyze the factors that affect the economic utilization of the collective attention of all parties involved, either actively or passively, in expertise sharing, and to use it as a guidance to design alternative communication mechanisms that have different cost-benefit structure by manipulating some variables in *CoCA* for different expertise sharing situations. By trying to change the number N , we devised the DML mechanism that is neither direct email nor mailing list nor BBS, but something in between email and mailing list with the feature of persistent storage of discussions. The comparison is not meant to rank the absolute superiority of communication mechanisms, but gives a clear understanding of each mechanism so that users and organizations can choose the most appropriate communication channel for their varied and nuanced communication needs in their specific socio-technical environment.

Among many systems that support expertise sharing [2, 12, 13], Answer Garden 2 [2] is most similar to the DML approach. Both approaches go through the expert identification and expert selection steps. They differ in the strategies of defining social relationships. DML defines social relationships based on inter-personal interaction histories while Answer Garden 2 uses organizational and physical proximities. The more important difference is the DML approach gives high priority to the individual preferences of experts, granting experts the full control of allocating their attentions with the introduction of socially aware communication mechanisms. The availability of choices and options helps the development of favorable attitudes toward expertise sharing [20], and this favorable attitude is critical for expertise sharing to become sustainable in an organization.

References

1. Ackerman, M.S., Halverson, C.: Sharing Expertise: The Next Step for Knowledge Management. Social Capital and Information Technology. MIT Press, Cambridge (2004) 333-354
2. Ackerman, M.S., McDonald, D.W.: Answer Garden 2: Merging Organizational Memory with Collaborative Help. Proceedings of CSCW96. (1996) 97-105
3. Ackerman, M.S., Piipek, V., Wulf, V.: Sharing Expertise: Beyond Knowledge Management. MIT Press, Cambridge (2002)
4. Arguello, J., Butler, B.S., Joyce, E., Kraut, R., Ling, K.S., Rosé, C., Wang, X.: Talk to Me: Foundations for Successful Individual-Group Interactions in Online Communities. Proceedings of CHI06. Montréal, Canada (2006) 959-968
5. Cross, R., Borgatti, S.P.: The Ties That Share: Relational Characteristics That Facilitate Information Seeking. In: Huysman, M., Wulf, V. (eds.): Social Capital and Information Technology. The MIT Press, Cambridge, MA (2004) 137-161

6. Dabbish, L.A., Kraut, R.: Controlling Interruptions: Awareness Displays and Social Motivation for Coordination. *Proceedings of CSCW2004* (2004) 182-191
7. Dieberger, A., Dourish, P., Höök, K., Resnick, P., Wexelblat, A.: Social Navigation: Techniques for Building More Usable Systems. *Interactions* **7** (2000) 36-45
8. Fussell, S.R., Kraut, R.E., Lerch, F.J., Scherlis, W.L., McNally, M.M., Cadiz, J.J.: Coordination, Overload and Team Performance: Effects of Team Communication Strategies. *Proceedings of CSCW98*, Seattle WA (1998) 275-284
9. Goldhaber, M.H.: The Attention Economy. *First Monday* **2** (1997)
10. Jackson, T., Dawson, R., Wilson, D.: The Cost of Email Interruption. *Journal of Systems and Information Technology* **5** (2001) 81-92
11. Lakhani, K.R., von Hippel, E.: How Open Source Software Works: Free User to User Assistance. *Research Policy* **32** (2003) 923-943
12. McDonald, D.W., Ackerman, M.S.: Expertise Recommender: A Flexible Recommendation System Architecture. *Proceedings of CSCW 2000* (2000) 101-120
13. Mockus, A., Herbsleb, J.: Expertise Browser: A Quantitative Approach to Identifying Expertise. *Proceedings of 2002 International Conference on Software Engineering* (2002) 503-512
14. O'Conaill, B., Frohlich, D.: Timespace in the Workplace: Dealing with Interruptions. *Proceedings of CHI95 Conference Companion* (1995) 262-263
15. Pentland, A.: Socially Aware Computation and Communication. *Computer* **38** (2005) 33-40
16. Rashid, A.M., Ling, K., Tassone, R.D., Resnick, P., Kraut, R.E., Reidl, J.: Motivating Participation by Displaying the Values of Contribution. *Proceedings of CHI2006* (2006)
17. Reder, S.: The Communication Economy of the Workgroup: Multi-Channel Genres of Communication. *Proceedings of CSCW1988*. ACM Press, New York (1988) 354-368
18. Reichling, T., Veith, M.: Expertise Sharing in a Heterogeneous Organizational Environment. *Proceedings of 9th European Conference on Computer-Supported Cooperative Network*. (2005) 325-345
19. Robillard, P.N.: The Role of Knowledge in Software Development. *CACM* **42** (1999) 87-92
20. Salancik, G.R., Pfeffer, J.: A Social Information Processing Approach to Job Attitudes and Task Design. *Administrative Science Quarterly* **23** (1978) 224-253
21. Schmidt, K.: The Critical Role of Workplace Studies in CSCW. In: Luff, P., Hindmarsh, J., Heath, C. (eds.): *Workplace Studies: Recovering Work Practice and Informing System Design*. Cambridge University Press (2000) 141-149
22. Simon, H.A.: *The Sciences of the Artificial*, 3rd edition. The MIT Press, Cambridge, MA (1996)
23. Szoestek, A.M., Markopoulos, P.: Factors Defining Face-To-Face Interruptions in the Office Environment. *Proceedings of CHI2006* 1379-1384
24. von Krogh, G., Spaeth, S., Lakhani, K.R.: Community, Joining, and Specialization in Open Source Software Innovation: A Case Study. *Research Policy* **32** (2003) 1217-1241
25. Ye, Y., Yamamoto, Y., Nakakoji, K.: A Socio-Technical Framework for Supporting Programmers. *Proceedings of 2007 ACM Symposium on Foundations of Software Engineering (FSE2007)* (2007) 351-360
26. Ye, Y., Yamamoto, Y., Nakakoji, K., Nishinaka, Y., Asada, M.: Searching the Library and Asking the Peers: Learning to Use Java APIs on Demand. In: Amaral, V., Veiga, L., Marcelino, L., Cunningham, H.C. (eds.): *Proceedings of 2007 International Conference on Principles and Practices of Programming in Java*. ACM Press, Lisbon, Portugal (2007) 41-50