A Framework for Supporting the Software Architecture Evaluation Process in Global Software Development

Muhammad Ali Babar
Lero, University of Limerick, Ireland
malibaba@lero.ie

Abstract

Software Architecture (SA) evaluation process requires a large number of stakeholders to be collocated for evaluation sessions. Given the increasing trend of using globally distributed software development teams, organizations are likely to be discouraged from introducing disciplined architectural evaluation practices that require stakeholders to be collocated. To address this issue, we propose that SA evaluation can be carried out using suitable groupware systems. In this paper, we present a framework for supporting the SA evaluation process using groupware systems. The framework highlights the changes required in the existing SA evaluation processes. We provide an illustrated example of modeling and mapping the activities of the proposed process on electronic workspaces provided by a groupware system. We have also identified some of the features a groupware system should have to successfully support the process.

Keyword: Software architecture evaluation, groupware support, global software development

1. Introduction

The software architecture community has developed several architecture evaluation methods such as the Architecture Tradeoff Analysis Method (ATAM) [13], Performance Assessment of Software Architecture (PASA) [45] and Architecture-Level Maintainability Analysis (ALMA) [31]. Most of these methods rely on the collaborative efforts of multiple stakeholders to perform various tasks. These include defining and refining business drivers, generating scenarios, and mapping the scenarios to the proposed architecture and reasoning about the architectural decisions. Most of these tasks are performed in Face-to-Face (F2F) meetings by co-locating stakeholders [28].

That means most of these methods overlook the fact that increasingly working teams in today’s organizations are distributed in the context of Global Software Development (GSD) [30]. Moreover, organizations are concerned about the time, costs and logistical problems involved in co-locating stakeholders [36]. That is why many organizations are implementing a virtual team strategy as the primary focus of their GSD policy.

Furthermore, architecture evaluation methods provide little support to address several issues that characterise F2F meetings, such as conformity pressures, dominating personalities, and cultural differences [34, 35]. We assert that collectively such issues may hinder the widespread adoption of software architecture evaluation practices.

In order to find suitable approaches to deal with these challenges, we suggest that groupware systems can provide a cost effective and efficient alternative to F2F meetings. Groupware comprises a diverse set of technologies that support collaboration, coordination, and communication among groups of people to improve productivity [35]. Groupware systems have been successfully used to support distributed teams for Requirements Engineering and Software Inspection. Studies have confirmed that computer-mediated processes provide promising ways to minimise meeting costs, maximise asynchronous work and preserve organisational resources [12, 16, 18, 19].

We propose that electronic workspaces based on groupware software can provide an appropriate mechanism for supporting the software architecture evaluation process in the context of GSD. We have developed a framework aimed at providing process and technical guidance about supporting the architecture evaluation process using electronic workspaces provided by groupware systems. This framework also identifies some of the unique tooling features required in order to support the software architecture evaluation process. Following section aims to describe the theoretical concepts upon which we have laid out the foundation of the framework reported in this paper.

2. Theoretical Background

2.1 The Architecture Evaluation Process

Abowd et. al. [1] proposed two broad categories of software architecture evaluation approaches: questioning and measurement. The former category includes techniques like scenarios, questionnaires, and checklists. The latter category consists of metrics and simulation. Scenario-based methods (such as ATAM, PASA and ALMA) are considered the most mature and well-known. Though there are differences among these methods [8], we have identified five common activities...
by comparing four main software architecture evaluation methods. These five activities make up a generic process of software architecture evaluation process [2], which we believe can be supported with a suitable groupware system. This process consists of following five activities:

1. **Plan architecture evaluation.** This is concerned with allocating organizational resources and setting goals for evaluation, selecting stakeholders, preparing inputs and deciding on the valuation team. This vital activity provides the roadmap of the process and identifies expected outcomes.

2. **Present and explain architecture.** During this activity, a software architect presents the architecture of the system under consideration. He/she also identifies the known architectural style or patterns used.

3. **Gather scenarios.** The purpose of this activity is to develop scenarios to characterize the quality attributes for a system. For instance the maintainability quality attribute can be specified by software change scenarios.

4. **Analyze architecture.** This activity aims to analyze architectural approaches using the developed scenarios. The most common approach is to map each scenario onto the architecture and reason about it.

5. **Prepare and manage results.** This activity is concerned with summarizing the results of all previous activities, interpreting the deliverables and presenting results to the sponsors.

One of the characteristics of architecture evaluation process is meeting-based activities. The requirement of holding meetings is partially created by the very nature of the scenario-based approaches. Gathering scenarios is an important activity aimed at eliciting quality goals of a system by generating general as well as concrete scenarios [9]. Stakeholders also prioritize the generated scenarios according to business goals. Architectural evaluation normally requires expertise and knowledge of different experts such as performance engineers and usability specialists. The affect of a particular quality attribute cannot be analyzed in isolation as quality attributes have positive or negative influences on each other, which may require tradeoffs between achieving different levels of different quality attributes. All these activities require group discussions and decision making processes, which necessitate meetings.

However, we argue that most of these activities do not necessarily need to be performed in a co-located arrangement. Rather, most of them can be done in asynchronous mode without affecting the quality of the outcome. The need for synchronous discussion can be supported by an electronic meeting system (EMS) [34].

### 2.2 Technology and Tasks

The study of group performance supported by technology has attracted great amount of research interest for a long time. It is said that group interaction and performance is greatly affected by the nature of and the level of difficulty of the task that a group is performing [33]. Moreover, influences of technology on group interaction and performance interact with task type. Thus, it is vital to understand the effects of different kinds of technologies on group task based on the nature of a task. There is a general consensus among group task researchers that differences in the nature of tasks should be taken into account when differences in group task performance are being studied [33].

Figure 1: McGrath’s group task circumplex [32].

Hence there have been several efforts to classify group tasks. It is not the aim of this section to provide a detailed discussion on various classifications of group tasks. Rather, it presents a brief discussion on two of the most influential theoretical concepts underpinning major research efforts in providing technological support for group tasks as these concepts have helped us understand the types of technologies required to support the process of distributed software architecture evaluation and provide the justification and rationale for the empirical studies we have been carrying out to assess the viability of the proposed process. These two theoretical concepts are: group task circumplex and task-technology fit [33].

McGrath’s task classification schema (shown in Figure 1) proposes that all group tasks can be categorized as one or another of four fundamental task performance processes, each with two main subtypes: to generate (ideas, plans); to choose (a correct answer); to resolve (conflicting views or interests); and to execute (in contests against another group). The four process are related to each other and are arranged in a circumplex structure (Figure 1) defined by two dimensions: (1) the kind and degree of interdependence among members in...
the task performance process (from cooperation to conflict or competition), and (2) the degree to which the processes involve conceptual versus behavioral activities. Features of each category are summarized in Table 1, which also shows that this research has classified the key tasks involved in different activities of software architecture evaluation process using the McGrath’s classification circumplex shown in Figure 1.

Table 1: Quadrants, task types, and brief explanation of the tasks according to McGrath’s task circumplex and software architecture evaluation tasks.

<table>
<thead>
<tr>
<th>Quadrants</th>
<th>Types</th>
<th>Nature of tasks</th>
<th>SA evaluation tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quadrant I</td>
<td>Generate</td>
<td>Planning tasks - goals setting, agenda setting, action-oriented plans</td>
<td>Plan analysis, Prepare documents, set goals</td>
</tr>
<tr>
<td></td>
<td>Type2</td>
<td>Creativity tasks - generate ideas, alternatives, and goals, brainstorming</td>
<td>Generate quality-sensitive scenarios</td>
</tr>
<tr>
<td>Quadrant II</td>
<td>Choose</td>
<td>Intelective tasks – aggregating and weighting preference, problem solving with correct answers</td>
<td>Evaluating architectural approaches, trade-off analysis</td>
</tr>
<tr>
<td></td>
<td>Type4</td>
<td>Decision-Making Tasks – Dealing with tasks for which agreed upon or preferred answer is the correct on</td>
<td>Choosing a suitable architectural approaches</td>
</tr>
<tr>
<td>Quadrant III</td>
<td>Negotiate</td>
<td>Cognitive tasks – Resolving conflicting views</td>
<td>Analysing and comparing architectural strategies</td>
</tr>
<tr>
<td></td>
<td>Type5</td>
<td>Mixed-Motive Tasks – Resolving conflicts of motivates</td>
<td>Prioritising scenarios</td>
</tr>
<tr>
<td></td>
<td>Type6</td>
<td>Performance/psycho-motor tasks – Psychomotor tasks performed against objective or absolute standards or excellence, excelling in something</td>
<td>Preparing and disseminating results by following set standards and agreed upon document structures.</td>
</tr>
<tr>
<td>Quadrant IV</td>
<td>Execute</td>
<td>Contests/battles – Competing tasks, resolving conflict of power</td>
<td>Prioritising scenarios, Justifying architectural decision</td>
</tr>
<tr>
<td></td>
<td>Type7</td>
<td>Performance/psycho-motor tasks – Psychomotor tasks performed against objective or absolute standards or excellence, excelling in something</td>
<td>Preparing and disseminating results by following set standards and agreed upon document structures.</td>
</tr>
</tbody>
</table>

Table 1 shows the tasks involved in different activities of software architecture evaluation process using the McGrath’s classification circumplex. The tasks are classified into four quadrants: Generate, Choose, Negotiate, and Execute. Each quadrant represents a different type of task, which can be represented as Type 1, Type 2, Type 3, and Type 4, respectively.

Throug there is no scientific justification for asserting as to why a certain task of architecture evaluation should be considered a type 1 task and not the type 2, it is asserted that this classification is largely correct as it is based on the researcher’s extensive experience of designing and implementing software architecture evaluation sessions for various industry partners and the researcher’s intimate knowledge of different methods of evaluating software architectures. From this classification of the key software architecture evaluation tasks, it is obvious that these tasks fall in all categories of McGrath’s task classification scheme. This classification of software architecture evaluation tasks combined with the task-media fit theory discussed next can help manager to select the most appropriate groupware medium for a particular task depending upon the amount of information required by a task and the capacity of a medium to provide that much information.

McGrath and Hollingshead used McGrath’s task classification circumplex model to provide another task classification based on the “information richness” concepts of the Media-Richness Theory (MRT) provided by Daft and Lengel, who posit that group tasks differ in how much transmission of information in terms of “content richness” is required by each task [14]. A simplified explanation for the “content richness” can be the amount of “extra” emotional, attitudinal, normative, and other meanings, beyond the literal senses of the symbols required to express a particular piece of the content. This “extra” information can be communicated perhaps via nonverbal and paraverbal channels. Such information adds to the richness of the information transmitted and provides basis for reducing the ambiguity of that information [33].

According to this second classification of tasks proposed by McGrath and Hollingshead based on the MRT, communication tasks can be classified into one or more of four types, in increasing order of complexity and information richness: tasks requiring groups to negotiate and resolve conflicting interests and views may require the transmission of the richest form of information, including not only facts, but also values, attitudes, effective messages and expectations. They also apply their theory to the study of groupware systems and propose a theoretical framework of task-technology fit based on information richness that needs to be provided by groupware systems for supporting various tasks [15].

Figure 2 illustrates the pattern of relations as follows: the best fitting combinations of information richness of task and technology lie near the main diagonal of the space. Contours that are successively distant from the diagonal represent less well-fitting combinations. Furthermore, assumptions are that task-technology combinations that depart from the best fit diagonal in: (1) the northeast direction (i.e. the technology provides more richness of information than the task requires) are poor fit with regard to efficiency and the communication medium brings distraction that is non-essential for effective task-performance and (2) the southwest direction (i.e. tasks that require more richness than the technology being used can deliver) are vulnerable to problems of effectiveness and quality of performance.

According to the task/media fit, a F2F meeting for generating scenarios may not be the most effective and efficient means of performing this task. Rather, a
computer-mediated meeting may be more suitable for such task. According to the framework, it looks like that the negotiation task for prioritizing scenarios might need a F2F meeting. Empirical studies have shown that even the quality of the output of a negotiation task, performed using a high quality audio/video facility, is as good as the output of a F2F meeting [15].

Moreover, research on socio-psychological effects in computer-mediated communication [26, 29] found that the reduction of socio-emotional exchange in less rich communication media enabled groups to pay less attention to the interpersonal aspects of the interaction and to actually focus more on the task-related matters. It is argued that such reduction in the influence of the socio-emotional feelings usually results in an increase in a group’s effort efficiency. It is asserted that several techniques of the evaluation methods are highly sensitive to socio-emotional feelings and organizational politics. For example, the results of prioritizing scenarios by raising hands can easily be affected by organizational politics, peer pressure, and conformance effect. That is why it is proposed that an anonymous voting system provided by a groupware system may be more suitable for objectively prioritizing scenarios.

2.3 Groupware Use for Software Development

It has been shown that F2F meetings for software inspections incur substantial cost and lengthen the development process [36]. Studies have called into question the value of F2F inspection meetings [38]. Studies have also indicated that computer tools may improve inspections [39]. Groupware-supported inspection processes have been successfully evaluated as a promising way to minimise meeting costs, maximise asynchronous work and conserve a number of precious organisational resources [18, 23]. Moreover, it has also been shown that the software inspection process can be improved with group process support [42]. The RE community has also been trying to devise and evaluate processes to improve the effectiveness and efficiency of its activities and to address the issues caused by an increasing trend of GSD. A number of studies have reported successful experiences in adopting groupware systems to improve RE [12, 16, 22].

These findings provide a strong motivation to systematically evaluate the pros and cons of using groupware applications to support the SA evaluation process in the context of GSD. Moreover, we are also interested in leveraging different groupware tools to improve the effectiveness and efficiency of the individuals participating in the SA evaluation process.


In this section, we present a framework for supporting the software architecture evaluation process using groupware systems in the context of GSD. Two of the core components of this framework are a generic software architecture evaluation process and electronic workspaces. We have already provided a brief description of the generic process of evaluating software architectures in Section 2.1. Here, we explain the concept and features of electronic workspaces that are being exploited in the reported research.

3.1 Electronic Workspaces

Virtual teams use virtual spaces provided by electronic workspaces (E-workspace). These virtual spaces are expected to create opportunities for the users to turn them into a place of collaboration [24]. Figure 3 shows the structure of a workspace to support a distributed SA evaluation process. All the objects of E-workspace are related with one another according to a model of collaboration presented in [11].

Actors assuming organizational roles collaborate with other roles within same workspace or another workspace to perform designated actions on artifacts [24]. Artifacts are either inputs or outputs for different actions. Roles are attached to a particular workspace and are unique within that workspace; more than one workspace may have the same roles attached to them. For example, a manager role may be attached to the architecture evaluation, planning and preparation workspace.

Each workspace has certain governance rules, which control the relationships among objects of that workspace. For example, a rule may provide read only access to SA documentation to the evaluation team, while the software architect has full access to that documentation. Each role can perform two types of actions on a workspace, namely individual actions...
(editing a document, sending a notification) or interaction actions (discussing ideas, prioritizing scenarios). E-workspaces also provide a wide range of notification features to support synchronous and asynchronous communication. Apart from providing tools to support collaboration, an E-workspace for SA evaluation also needs to contain tools that can help build and manage organizational memory and a repository of reusable artifacts. It should also support activities like idea generation and organization, and decision making by evaluating alternatives available.

![Figure 3: A structure of electronic workspace and its major components](image)

### 3.2 Groupware-Supported Evaluation Process

We have mentioned that most of the activities of the SA evaluation process are performed in F2F meetings [13, 31]. Such meetings can create logistical problems such as scheduling difficulties, traveling hassles, conformity pressure, and so on [34]. Our research aims to design and evaluate an effective and efficient way of involving physically distributed stakeholders in the architecting processes without making them travel, while improving the overall process.

We assert that the activities of the SA evaluation process can be performed using groupware systems. Instead of requiring the stakeholders to be collocated at a physical space, E-workspaces are used to overcome the limitations of the existing architecture evaluation approaches in the context of GSD. Web-based groupware systems have played a vital role in improving the processes in different disciplines by minimizing dysfunctional behavior and enhancing group productivity [20, 21, 34, 37]. These findings of using groupware provide encouragement and guidance for the development of our groupware-based process.

Figure 4 shows a model of the groupware-supported SA evaluation process. It shows the five activities of the generic process and different tools that an E-workspace should have to support those activities. The links between different activities and tools represents the fact that each activity of the process can use one or more tools to carry out the tasks involved in a particular activity. In the following, we briefly explain how groupware support can improve the process by providing several technical and non-technical benefits. Our process is expected to help manage the evaluation process in a disciplined manner.

The evaluation planning and preparation activity aims to prepare the roadmap of the evaluation process, getting the inputs ready, and setting the evaluation criteria for the outputs. Groupware support is expected to automate several tasks by helping with identifying and checking the availability of the evaluators and stakeholders based on predefined criteria, suggesting evaluation methods, assigning the tasks and notifying the participants. Groupware support enables a manager to optimize the use of the available resources [20]. This process easily accommodates changes by responding to the needs of any unanticipated development [24].

The proposed process is expected to benefit the architecture presentation activity in several ways. For example, the need for having lengthy architecture presentation meetings will be minimized as such meeting and subsequent discussion is conducted using different collaborative tools (e.g. drawing tools, EMs, chat rooms). Stakeholders can see different views of the architecture on their screens and raise their concerns in synchronous or asynchronous discussions. Moreover, the process is expected to provide a knowledge repository to store and retrieve the design decisions, known patterns and their effects on the desired quality attributes. Online availability of such knowledge can greatly facilitate the stakeholders to fully comprehend the architectural approaches being used and then raise their concerns with informed arguments. We have developed such a knowledge repository [3] to enable the distributed stakeholders of the software architecture process to capture and sharing architectural knowledge [4]. This process will also benefit from integrating groupware systems and Computer Aided Software Engineering (CASE) tools [22] to enable different classes of stakeholders view and modify architectures during discussion.

Scenario elicitation and prioritization is the most expensive and time consuming activity. Our research program particularly aims to improve this activity by developing different techniques to help enhance the quality and quantity of the generated scenarios with minimum resources. That is why we have begun our
empirical investigation to assess the effectiveness of the groupware support with scenario development activity [5]. Groupware systems provide different tools for brainstorming, organizing, and prioritizing quality sensitive scenarios in distributed arrangement [34].

Our process uses prepared group method of developing scenarios [10] along with both synchronous and asynchronous communication modes. For example, stakeholders can develop and prioritize individual scenarios using a tool. All the stakeholders can see each others’ scenarios, make comments and discuss each scenario using a discussion board or chat room. Then they can use an EMS to integrate their scenarios and brainstorm new ones [35].

Figure 4: A model of a generic process of architecture evaluating supported by E-Workspace.

An evaluator can encourage them to diverge from familiar thinking patterns using a brainstorming tool. A divergent group can be stimulated towards more focused thinking using an idea organizer [44]. Instead of prioritizing scenarios by raising hands or openly assigning votes to certain scenarios [13], our process will take the advantage of a sophisticated voting tool that keeps the process anonymous, which is important to separate ideas from organizational politics [34].

Existing approaches have not paid any consideration to social and political issues. Thus, we believe that a groupware-supported process will also address a number of socio-technical issues (i.e. egoism, unfair floor control, and deficiency in a spoken language) [28]. Moreover, scenarios and other information are currently captured using flip cards or a “chauffeur-driven” computing model; both are inadequate to effectively and efficiently assimilate and process the huge amount of information generated during scenario development workshops. Several studies have proved that groupware systems can provide an effective mechanism of capturing and processing large amounts of information [20, 21, 34], which can be instantly disseminated. We anticipate similar benefits of introducing groupware support.

The groupware-supported process is also expected to greatly facilitate the task of analyzing architectural approaches. Having performed this activity in physical meetings, our conclusion is that this activity is the least suitable for a F2F meeting of large number of stakeholders. We have found that more than a meeting, this activity needs an effective and efficient mechanism of sharing information (with human or computers), finding and evaluating design alternatives, identifying risks and non-risks, making tradeoffs, and storing and retrieving design rationale. We realize that a single application may not be able to support all these tasks as most of the decision support tools tend to focus on a very small group of tasks [27]. However, an evaluator can use a range of decision support tools and discuss the finding within the groupware environment. Our groupware-supported process will also provide a design decision repository to facilitate the decision making process by presenting architecturally significant information in a readily usable format [3].

This process will greatly improve the task of interpreting the evaluation findings and preparing reports by providing online document management and group memory tools. The evaluation team can search the annotated information to discuss and clarify ambiguities and debatable points or contact the source of a particular scenario or architectural concern using collaborative tools. The stakeholders can also see the evaluation report online and leave comments. Findings are electronically disseminated. Post-evaluation feedback tasks are improved by using electronic forms, reminders, and getting the responses directly entered into a database or placed on a Wiki.

A long term benefit of this approach is the possibility to evaluate the performance of the SA evaluation process with a wide variety of data (such as scenarios, architectural approaches, sensitivity points) along with its available meta-data, which can be processed to develop different metrics. One of our main goals is to design and validate some mechanism of quantifying the benefits of performing evaluations. We believe that providing a tool to capture and analyze the data used or generated during the evaluation process can be the first step towards that goal.

3.3 An Illustrated Example

In this section, we illustrate the use of E-workspaces for the distributed architecture evaluation
process by designing some of the activities, inputs, outputs, and roles involved in the evaluation process.

For designing the process structure of the distributed software architecture evaluation process suitable to be implemented on workspace-based systems, we use a diagrammatic notation, which is a modified version of Rich Picture modeling language [24]. According to this notation, activities, artifacts, and roles are represented by the clouds, rectangles, and human figures respectively. The arrows linking artifacts and activities represent inputs to and output from the activities. While the lines linking roles and activities represent their relationships. A rich picture representation of the architecture evaluation process is shown in Figure 5. It shows main roles, artifacts, and their interaction with different evaluation activities. The artifacts are either inputs (e.g. business goals and architectural requirements) or outputs (e.g. prioritized scenarios, risks and non-risks) of different activities.

![Figure 5: Rich picture showing a model of workspaces for the SA evaluation process.](image)

Each of the activities and participants of a workspace are containers of knowledge used/generated during each of the activity shown in Figure 5. The participants of each workspace take on their respective roles (e.g. architect or business manager) to perform individual as well as group tasks. For example, the rich picture shows that a participant may take on the role of a manager to perform the architecture evaluation planning and preparation activity. This activity needs artifacts like business goals and project plan as input and produces or modifies artifacts like software requirements specifications (SRS), software architecture documentation, and the evaluation plan as the outputs, which are used as the inputs for other activities, e.g., present and explain architecture, gather scenarios, and analyzing architectural approaches used in a proposed software architecture. The architecture evaluation process modeled using rich picture can be implemented on a system that supports the workspace paradigm ((e.g., systems reported by Hawryszkiewycz [25] and Stevens et al. [41]). The mapping follows simple rules, namely:

1. For each activity of rich picture create a top-level workspace.
2. Roles associated to a particular activity becomes roles on the workspace for that activity.
3. Artifacts consumed or produced by an activity become artifacts on the workspace of that activity.

Figure 6 shows the workspaces, roles, participants, and artifacts created in LiveNet to illustrate the mapping between the models of the software architecture evaluation process presented in Figure 5. Each of the high-level activity of the architecture evaluation process shown in Figure 6 (i.e., Plan architecture evaluation, Present and explain architecture, Gather quality scenarios, Analyze architecture, and reporting) has its own workspace that can be entered by clicking on the respective activity. Figure 6 also shows the artefacts, roles, and participants involved in the process. Each of the workspace can be configured based on the requirements of a particular evaluation session. Different tasks can be supported by external tools on these workspaces by providing links to those tools.
Figure 6: An electronic workshop for generating scenarios activity of the architecture evaluation process. Once the work progresses, an evaluation manager may need to accommodate new process requirements that require addition of new workspaces, artifacts, or roles, or delete existing ones. Moreover, these workspaces only support the general evaluation process presented in this paper. Different processes will have different workspaces.

4. Features Required

Based on the findings from our empirical studies [5-7] conducted to evaluate the different aspects of the proposed framework, anecdotal evidence gathered from our experiences of designing and evaluating architectures, our extensive discussions with practitioners and literature review, we have identified a preliminary set of features required of a support environment for the proposed process. Apart from the general collaborative tools (such as synchronous and asynchronous communication mechanisms, document management), a groupware system should have following features to successfully support the SA evaluation process in the context of GSD:

Workspaces with peer-to-peer to communication – workspaces must support communication among objects (artifacts, activities, roles) within a workspace as well as between different workspaces. There must be strong correspondence between different workspaces.

Modeling support – SA is documented using different modeling languages. We need a tool that can either provide an integrated modeling tool or can easily import and export diagrams of the SA being reviewed from several CASE tools.

Rationale management repository – our research efforts are concentrated on enabling organizations to do more with less in SA design and evaluation. A sophisticated rationale management repository is vital to this end. Such a repository improves the reusability of architectural information (such as scenarios, patterns, tactics, design alternatives) [13]. Availability of architecturally significant information annotated with design rationale in a ready to use format can enhance the quality of design decisions and evaluation results [3].

Concurrency control – since we expect a number of stakeholders to be working concurrently, a good concurrency control mechanism is vital. The system must resolve conflicting requests, be highly responsive and robust, support data replication and turn taking, and not be a hindrance in tightly coupled teamwork [17].

Evaluation measures – tool must be able to analyze the data consumed or generated during an evaluation and build basic metrics to quantify costs and benefits. For example, the ratio of reused and new objects consumed in different types of evaluations, number of participants and quantity and quality of scenarios generated, number of design decision made and alternatives considered for each decision, and so on.

Spatial Hypertext Wikis – Wikis are effective mechanism for sharing knowledge in software engineering. The advantage of using Wikis for architectural knowledge management (AKM) is that architecture development can be performed in a collaborative, distributed and a reusable way. In order to support the AKM in the proposed process, we follow Solis and Ali [40] in combining Spatial Hypertext and Wiki technologies in the manner described in the tool called, ShyWiki, which provides richer environment for sharing and visualizing architectural knowledge.

5. Empirical Evaluation

We have designed an empirical research program to evaluate the effectiveness of the proposed SA evaluation process. Instead of evaluating the whole SA evaluation process in a distributed environment, we have begun our empirical investigations with the most expensive activity of the SA evaluation, namely developing scenarios. We have run two controlled experiments to compare the performance of collocated and distributed groups based on the quality of scenario profiles developed. For our study, a “collocated” group is one in which the participants meet face-to-face to develop scenario profiles, and a “distributed” group is one in which participants use the electronic workspaces.
provided by a groupware to develop scenario profiles. By comparing the quality of the scenario profiles of collocated groups with those of distributed groups, the experiments attempted to tease out the effects of the meeting arrangements, face-to-face or distributed, that may have on the quality of the artifacts.

The analysis of the quantitative data gathered from both experiments suggested that the effect of a distributed meeting is better than a F2F meeting in terms of the quality of the scenario profiles developed. For detailed results, see [5, 6]. However, the analysis of the data gathered through post-experiment questionnaire in both studies revealed interesting trends. It was found that the majority of the participants believed that their individual and group performances were better in collocated arrangement, which is contrary to the findings based on the quantitative analysis. That means we did not find any significant support for the groupware-supported distributed arrangement a large majority of the participants of the both studies was extensively using Internet based collaborative tools (e.g. NetMeeting, Yahoo Messenger) for professional and personal purposes. The participants provided a number of reasons for disliking the tool-based arrangement, such as a lack of body language, collocated being more natural and conventional, typing problems, slow collaboration, time lag in communication, and so on. We also used a psychometric based questionnaire for gathering data, which is being analyzed to discover socio-psychological aspects of using groupware systems to support the architecture evaluation in a distributed arrangement.

6. Conclusion and Future Work

Currently, SA evaluation is usually performed in a F2F meeting. Collocating large numbers of stakeholders is difficult and expensive to organize, particularly in the context of GSD projects. We have proposed a groupware-supported SA evaluation process, which does not require the simultaneous participation of all the stakeholders. Nor do stakeholders need to be physically co-located. The process is intended to address a number of logistical issues that characterize the SA evaluation approaches by taking advantage of the collaborative and social networking technologies, which have solved similar problems in other disciplines [12, 18].

Many of the identified benefits of introducing groupware systems await rigorous assessment in the context of architecting processes. We also realize that any technological initiative without considering group wellbeing and member support is doomed to failure [37]. Although, social and organizational aspects are not within the scope of this paper, our ultimate intent is to evaluate our solution in terms of the quality of the deliverables, the resources saved, and the utility to groups and organizations [17, 18].

We have been using experimentation to evaluate the effectiveness of the proposed solution and identify the features that a groupware application should have to successfully support the process. For this purpose, we have conducted two experiments. The findings of these experiments support our assertion that software architecture evaluation process can be supported using groupware systems without compromising the quality of the output. We intend to conduct further studies to assess the viability of the other activities of the process. We are also developing some of the tools that are expected to be required for the proposed process. We have already developed a knowledge repository [3, 43], which can be integrated into a collaborative application like LiveNet in order to make the workspaces of such an application more suitable for supporting the process.

Acknowledgement

This work is partially supported by Science Foundation Ireland under grant number 03/CE2/I303-1.

7. References