Modeling Virtual Meetings within Software Engineering Environment

Dr. Aiman Turani
Applied Science University
Amman, Jordan
aimant@asu.edu.jo

Dr. Ayman A. AbuBaker
Applied Science University
Amman, Jordan
a_abubaker@asu.edu.jo

Dr. Faiz Al-Shrouf
Amman Arab University
Amman, Jordan
faiz_alshrouf@hotmail.com

***Abstract*—It is a common scenario to see project's stakeholders, such as managers, team leaders, and developers carrying out their meeting in the online environment without a suitable preparation and facilitation For instance, stakeholders engaging in negotiation sessions trying to communicate system requirements in the virtual environment might face requirements misunderstanding which in turn might cause a whole project to fail. Usually a meeting agenda and design is implicit in the facilitator's head. Conducting such meetings without obvious structure would potentially lead to various problems such as no one seemed to be in charge? Or there was no clear reason to meet or no agenda etc.**

In this paper, we are presenting a general framework to model group-based activities and meetings within software engineering field in a simplified and formal manner. Traditionally, managers submit their web-based group meeting information in a form of text-based instructions [1]. Then a group facilitator or chairperson will lead the group throughout the meeting to achieve the desired objectives.

These types of meeting are relatively easy to manage in face-to-face environment where web based meeting in the other hand, is more challenging to facilitate and manage. Therefore, more and more specialized tools are immerging to manage and facilitate such meetings. For instance, Adobe Connect [2] is tool for facilitating web-based meetings. These tools usually allow facilitators to organize and prepare the meeting floor by inserting specific collaboration components such as chat, whiteboard, voting, etc. Then during the meeting the facilitator guides participants using the video or text component. This usually leads to undesirable outputs due to the lack of a clear structure or agenda in addition to the virtual distance that weakens the communication.

In this paper a two level of modeling views are proposed, the static view and the dynamic view. The static view mainly represents the aggregation association between session stages, phases, and activities. The dynamic view shows how meeting modules or elements within various activities are accessed during runtime. A particular modeling tool called "MEET-Represent" is being developed according to the proposed process in this paper to assist meetings designers to collaborate and model their meetings in advance to ensure the coherence and alignment of the meeting mini-tasks with the overall meeting objectives and goals.

Keywords-Software Engineering, Virtual Groups, Modelling, Meeting Design.

I. INTRODUCTION

As known, during software development there are many activities that need to be done by the project's stakeholders. These activities are typically carried out in a sequential or parallel form [3] during the project's life time cycle. A wide range of activities could be found. An activity could be reading software development artifacts, engaging in assigned individual task, or participating in various group-based meetings such as brainstorming, voting, debating, etc. These group meeting are sometimes costly when done in face-to-face mode and ineffective when performed on the online environment.

Modeling has its roots in major disciplines, such as Architecture Engineering, Mechanical Engineering, Software Engineering, etc [4]. Representing a system using graphical notations has been adopted in many disciplines to simplify the description of a system while preserving only a limited number of its original details [5]. This is usually done before the implementation of a system for better understanding and higher quality. Nevertheless, there has been a little evidence that managers within software engineering field are following a proper modeling method while designing their meetings especially within the online environment. These meetings are reporting serious problems such as, keeping participants focus on meeting agenda, proceeding according to a specified time table, communication collagen where more than one person try to talk at the same time causing more delays, jumping in and out during the meeting causing further frustration, etc.

II. MODELLING WITHIN THE MEETING DESIGN PHASE

Meetings preparation and organization within software engineering discipline is considered to be a complicated task due to its diversity nature [6]. It may include many interconnected tasks, such as defining meeting objectives, group nature, assessment types, individual and group activities, etc. Designing a virtual meeting is an even more challenging task. In one hand, the design should relate to a wide spectrum of activities and tools performed and used in a virtual distance, and in other hand it should be done in a simplified and systematic form. Typically, modeling is a core component within design phases. Modeling should represent the logical relationships between various entities. In addition, modeling notations should be limited to a set of few numbers, to make it easier for managers and designer to remember and use. Also, different modeling views should be used. No single view could present every design aspect or behavior. Finally, modeling should be abstract where only important concepts should be represented.

In this paper we are introducing a general modeling process as shown in Figure 1. In this process a meeting's creator needs to define the session objectives. Then he needs to state his design specifications, such as group members, group activities, roles, assignments, etc. After that the meeting designer, which could be the manager himself, if he has the required knowledge, will analyze these specifications and create a multiple modeling views. These views could be used as guideline during the meeting time. Also specialized applications could implement these designs by realizing these models and diagrams into appropriate supporting tools during the meeting.

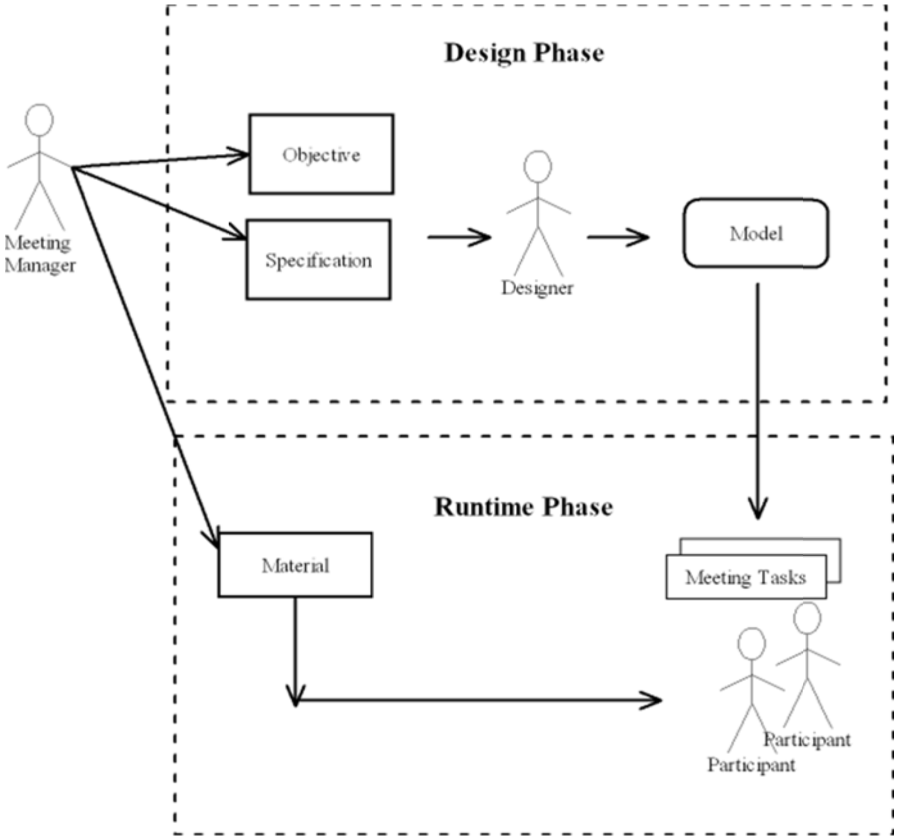


Figure 1: E-Meeting Design Framework

III. MEET-Represent

MEET-Represent is a graphical-based modeling tool. This tool is being built according to the framework that has been proposed in this paper.

Currently *MEET-Represent* includes two types of views along with a small set of graphical notations. The two types of views: Static view and Dynamic view. The Static view represents the relations between various session entities or nodes. The relations' type is mainly the "include" relation that represents the hierarchal nature. The Dynamic View describes primary the paths and conditions between various session nodes.

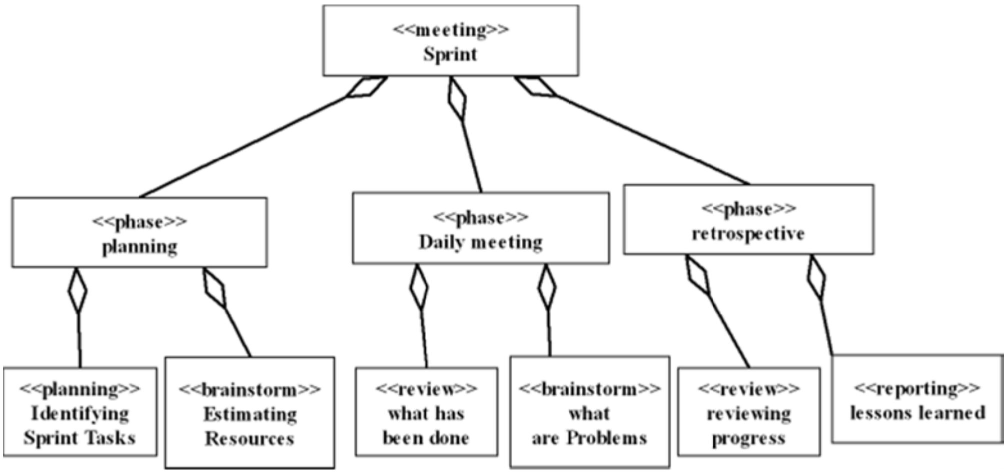
A. *Static View*

The Static view represents the aggregation association between session nodes at different levels of granularity. At a higher level, a session is composed from different stages, such as, pre-session, post-session, etc. A stage is composed from different phases where each phase may include various resources and activities.

Figure 2 shows a Static View example where a Sprint meeting within the Agile developments process [7] is modeled. Usually, each Sprint is preceded by a planning meeting. In this meeting, tasks for the Sprint are identified and estimated and also the Sprint goals are made. At the end of Sprint, a review or retrospective meeting is usually done where the progress is reviewed and lessons for the next Sprint are identified.

As shown in

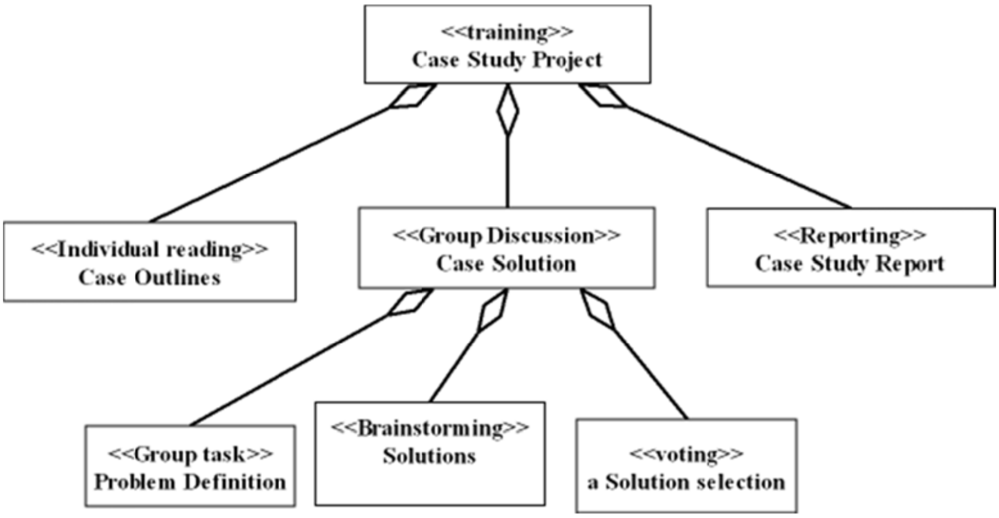
Figure 2, each phase is assigned to specific tasks. The meeting phase stereo-type could be "Planning", "Retrospective", etc.



<<meeting>> Sprint
<<phase>> planning
<<phase>> Daily meeting
<<phase>> retrospective
<<planning>> Identifying Sprint Tasks
<<brainstorm>> Estimating Resources
<<review>> what has been done
<<brainstorm>> what are Problems
<<review>> reviewing progress
<<reporting>> lessons learned

Figure 2: Modeling Sprint meeting Static View

Figure 3 shows another example where it represents a traditional Case-Study meeting view. As shown, it can be detailed in a multilevel layer according to the designer's desire.



<<training>> Case Study Project
<<Individual reading>> Case Outlines
<<Group Discussion>> Case Solution
<<Reporting>> Case Study Report
<<Group task>> Problem Definition
<<Brainstorming>> Solutions
<<voting>> a Solution selection

Figure 3: Activity phase Static Diagram

A detailed Use Case that explains how a meeting designer creates a traditional meeting static view within *MEET-Represent* is demonstrated in TABLE I.

TABLE I. Building a "Meeting Static View" Use Case

Use Case	**Creating a traditional Meeting Session Static View**
Actors	Manager, Designer
Precondition	A manager logged into the system
Success End condition	The manager has successfully create a meeting session static view
Main Flow	1. The manager chooses from menu Create New Meeting Static View. 2. The system inquires the manager to fill in session information (Session Name, Session Objects, and Session Description). 3. The system presents a blank page including the session box in the middle and a toolbar on the side to enable meeting designers to draw their designs. 4. The Designer starts by dragging session's components to the stage. 5. The system inquires manager to fill in information related to this component (Objective, Description, Duration, Participants, Roles, etc) 6. The manager can detail these components further by creating and linking children component /objects using aggregation association. 7. For creating more session's components, manager repeats steps 4-6. 8. The manager saves this design form by clicking on the save icon in the menu.

Design Reusability can be archived by exporting and importing design details. *Meet-Represent* can generate automatically the session's schema representing all elements and details included in the static view.

B. *Dynamic View*

Dynamic view shows how meeting modules or elements are accessed during runtime. The most common scenario is the linear style where participants finish one meeting component before moving to the next. The other type is the parallel style where more than one component is done concurrently.

MEET-Represent provides limited customization options for meeting designers. There are two common types of customization. The first type is the "Open-form", where participants can access any component within the session. Secondly the "Restricted-form" where participants can access certain component only if a specific condition holds such as specific time, completion of other module, or facilitator permission.

The scenario for creating a dynamic view is presented in TABLE II. Any meeting phase that includes children tasks can have a dynamic view. If the designer did not specify the dynamic view, the system assumes the default view in which participants can access any task at any time without any restrictions.

TABLE II. Creating a "Meeting Dynamic View" Use Case

Use Case	**Creating a Meeting Dynamic View**
Actors	Designer
Precondition	• A designer logged into the system • A Meeting Static View was built
Success End condition	The designer has successfully build a meeting dynamic view
Main Flow	1. The designer right-clicks within a meeting static view. 2. Chooses from sub-menu create a dynamic view. 3. The system presents on the sidebar all the children meeting tasks within the selected meeting phase. 4. The designer starts by dragging meeting tasks to the stage. 5. The designer connects these tasks using a link association. 6. The designer saves this design form by clicking on the save icon in the menu.
Alternative Flow	5-a The designer can specify a precondition regarding accessing meeting task by selecting an accessing rule. 6-a if not all meeting tasks are presented in the diagram, the system will present a warning message.

IV. CONCLUSIONS

The main objective behind this research was to define a modeling scheme that enables meeting managers to effectively engage in designing effective e-meetings within software development environment. Assuming that providing the most advanced communication tools will ensure a successful virtual meeting is a mistaken belief where many organizational managers are reporting serious frustration when conducting virtual informal meetings with their employees. There are usually the risks that participants cannot start, get lost, or cannot reach the stated meeting's objectives.

The process of designing and modeling is important specially in developing large-scale software products. We are developing a tool "*MEET-Represent*" as a proof of concept of such methodology. *MEET-Represent* supports multilevel of granularity, multiple views, Static and Dynamic view, and a high learnability curve where very few graphical notations are need. An early phase of evaluation indicates promising results such as better retention and faster tasks fulfillment. A full evaluation study will be conducted to verify targeted goals and objectives of this research. .

ACKNOWLEDGMENT

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the financial support granted to cover the publication fee of this research article.

REFERENCES

[1] Sammi Chan, Alicia Cuello, Brian Kirk, Lara Procknow, Joy Smesny, Paul Trinh, "Web-Based Training" , *Topic Report developed by Wee-Go-Commando Group, 2013*, from http://coursesite.uhcl.edu/BPA/bpa/training/Web-BasedTraining.doc

[2] Adobe. Retrieved 22 January 2014, from http://www.adobe.com/products/breeze/index.html.

[3] Fan, Shaokun, " Electronic Dissertation," University of Arizona, 2013, from http://arizona.openrepository.com/arizona/handle/10150/242374?mode=full&submit_simple=Show+full+item+record

[4] MAHONEY, MICHAEL S, " FINDING A HISTORY FOR SOFTWARE ENGINEERING ," *ANNALS OF THE HISTORY OF COMPUTING, IEEE*, VOL. 26, 2004, PP. 8-19.

[5] R. Koper, "Modeling unit of study from a pedagogical perspective the pedagogical meta-model behind EML." Retrieved January 2014, from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html.

[6] Damian, D.E, " The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization ", Requirements Engineering, 2002. Proceedings. IEEE, pp. 319 - 328.

[7] Schwaber, Ken; Beedle, Mike. *Agile software development with Scrum*. Prentice Hall. ISBN 0-13-067634-9, 2002.

AUTHORS PROFILE

Aiman Turani received the BSc degree in electrical/ computer engineering from Louisiana Tech University, Louisiana, USA, in 1990, MSc degree in electrical/computer engineering from Louisiana Tech University, Louisiana, USA, in 1992. He finished his PhD from the Information Engineering School, Sydney University, in 2007. His research interests include synchronous collaborative and learning designs methods. He is currently working at the IT College of Electrical and Computer Engineering, Jordan.

Faiz Al-Shrouf received his Bsc from KSA University in Computer Science, Msc in Mathematical Statistics from Yarmouk University, and PhD in Software Engineering from University Science of Malaysia. He is currently working as an Assistant Professor at Faculty of IT at Amman Arab University/Jordan. His main research interests: AgentOriented Engineering, Mobile Agents, Software Design Patterns, and MCommerce.

Dr. Ayman A. Abubaker: Asst. Prof. at Electrical and Computer Engineering Department, Applied Science Private University, Amman-Jordan. He got his PhD in Electronic Imaging and Media Communications (EIMC) from University of Bradford-UK. His main interests are image processing, machine learning, and intelligent mobile robots.