Modern Meeting Management and Information Retrieval

Automatic Protocol Generation and Meeting Workflow Support

Jennifer Schiller
Technische Universitaet Muenchen, Siemens AG
schiller@in.tum.de

Abstract

Communication is a key component in the daily software project management. It is the quintessential carrier of semantic information. But speech is transient and with it the information exchanged in meetings.
To organize a successful meeting, some rules have to be followed. One is to summarize important - said - information in a written protocol to externalize the knowledge shared in the meeting.
The act of manually writing meeting minutes implicate a lot of problems. The protocol is published too late, inconsistent, or incomplete, due to the limited perceptivity and time of the protocol writer. Hence, we focus on an approach to automatically generate protocols and make them accessible to the right person. Therefore, keyword spotting, based on a predefined and meeting-type dependent grammar, is applied to meeting recordings and the discussed information is extracted and accordingly pipelined into the project information workflow.

This paper presents a lightweight framework of an information and knowledge management approach to capture communication in meetings.

Categories and Subject Descriptors D.2.9 [*Software Engineering*]: Management - Life Cycle

General Terms Management, Documentation, Human Factors, Algorithms

Keywords Meeting, Communication, Information Retrieval, IBIS, Speech Recognition, Workflow Support

1. Introduction

Meetings and verbal collaboration have become an integral part of the workaday life. In Meetings, the current status as well as the project progress is discussed; over problems, decisions and changes is talked.

Speech is a key component in the project management life. It is the quintessential carrier of semantic information. But communication is transient and with it the information discussed in a meeting.
In the area of meeting management a lot of „best practices“and proceedings were published in the past, that describe how to prepare and execute a successful meeting. Also Issue Based Information Systems (IBIS) [1], invented by Werner Kunz and Horst Rittel in 1970, is a method used to support group discussions. IBIS are meant to help coordinating and planning decision processes.

Experiences and theories from the past are still applied in modern meeting management, however there are new challenges. The number of meetings today is rising more and more, for example. Especially the emergence of agile methods and practices [2, 3] and their increasing acceptance in the software development environment promote the unwritten communication, e.g. by a Daily Scrum Meeting [4]. However, when there are no more documents, an important knowledge management source gets lost, as communication is volatile. Also traceability of information is impeded. In this context, meeting management literature demands the creation of meeting minutes and protocols, but the magnitude of meetings per day makes it impossible to write all the protocols.
The obvious solution, beside the creation of manual protocols, was to record all meetings and save the audio files. First evaluations of our projects [5, 6] admittedly showed that a pure recording of meetings is not enough. The retrieval of specific information as well as the search through the audio-database is painful and often not fruitful. A post-processing of the audio files is indispensable.

OOPSLA 2009, October 25–29, 2009, Orlando, Florida, USA.

Aim of our approach is to provide a framework to support meeting workflows, comprising post-processing, information retrieval and acquisition of information based on meeting recordings. The framework realizes the ideas of IBIS and will embrace the traditional meeting management approaches. The idea is to support the whole meeting workflow: from a meeting, its recording to the point of extraction of important information out of the meeting and its appropriate storage and creation of reports. Therefore, a meeting taxonomy with a dedicated meeting grammar is defined, which allows our tool to find and extract important information from an audio file and save them appropriately.

This paper gives an overview of our framework, introduces our meeting taxonomy, and shows the possible application in a software development project.

2. Related Work and Innovative Ideas

Meetings are exercised from time immemorial. In a meeting, two or more people come together for the purpose of discussing a (usually) predetermined topic. „We are a meeting society - a world made up of small groups that come together to share information, plan, solve problems, criticize or praise, make new decision or find out what went wrong with old ones. “[7]
In organizations, meetings are an important vehicle for personal contact. They are so common and pervasive in organizations, that many take them for granted and forget that, unless properly planned and executed, meetings can be a waste of time and resources.

Most meetings have their roots in some version of parliamentary procedure, where the manager sits at the head of a conference table, controlling the dicussion, while a secretary takes minutes. Parlamentary procedure is fine for formal debate, but it is not at all suited to solving problems more or less informally, collaborating, working together to reach agreement, and coping with complex, interdependent issues. So it‘s no wonder meetings generally don‘t work very well. The environment has grown vastly more complex in the last half-century. [7]
For this reason, a lot of best practices, advices and optimal meeting procedure have been released. These comprise, having a meeting facilitator, but also to plan the structure of a group session including the purpose, desired outcomes, and procedure, summarized in an agenda. „Mining the group gold“[8] involves also a meeting recorder and writing a protocol to keep the decisions, brought up problems, etc.

Beyond that, the idea of Issue Based Information Systems (IBIS) provides a structured procedure and a rhetorical model for meetings to enhance their outcome. IBIS is designed to support, to document, and to coordinate information processes. It guides the identification, structuring, and settling of issues raised by problem-solving groups, and provides information pertinent to the discourse. The concept of IBIS rests on a model of problem solving by cooperatives as an argumentative process. About an initially unstructured problem area or topic a discourse develops. Issues are brought up and disputed because different positions are assumed. Arguments are constructed in defense of or against the different positions until the issue is settled by convincing the opponents or decided by a formal decision procedure.
Elements of the system are topics, issues, questions of fact, positions, arguments, and model problems. Issues are the organizational „atoms“of IBIS-type systems.
IBIS is also a documentation and reporting system which permits fast and reliable information on the state of discourse at any time. These representation of the various relations between issues, questions, etc., by graphic display of the state of argument is called an Issue Map.
The Issue Map, 1970 style, involved pen and paper. But over the following years, IBIS was refined and further developed. Many of the components of the 1970s version were made obsolete by advances in technology and business practices, but Issue Maps in particular, remained. In the 1980s, the era of personal computers dawned and later pioneers such as Jeff Conklin, could see the potential that IBIS had by utilising a computer-based visual display. Independently from Rittel, Conklin was pursuing ways of capturing design rationale. He adapted IBIS for use in software engineering and created the gIBIS (graphical IBIS) hypertext system.

These days, a skilled IBIS practitioner can make a massive difference in helping a group develop a shared understanding and commitment of a problem. IBIS, at its heart, is a language specifically designed to break down the often convoluted and complex structure of a conversation into something much more simple to understand and digest. The premise of IBIS is that no matter how complex or argumentative an issue is, we can break it all down to just three basic artifacts: Questions, Ideas, Arguments (pros and cons).
Ideas respond to questions, offering possible solutions to the question. The arguments argue for and against the various ideas. Questions can then be expanded on or challenge other questions, ideas, or arguments.

Issue-Based Information Systems are used as a means of widening the coverage of a problem, by encouraging a greater degree of perspectives. Additionally, it helps to make the design process transparent. Transparency here refers to the ability of observers as well as participants to trace back the process of decision-making.

It seems that, based on the realization of the meeting managment advices and the application of an Issue Based Infor-

mation System the meeting process should be perfect. But the increasing number of meetings entail new challenges, chances and problems, which need new approaches.
In the following, we will focus on the documentation of meetings and protocol generation, as there are already a lot of tools (e.g. [9]) for the creation of agendas. Especially a solution for ad-hoc meetings will be presented. Traditionally, meeting minutes have to be written manually. This holds a lot of problems just now, where many protocols should be created. The minute taker can‘t keep up writing and listening. So he will probably miss the one or other statement. Additionally, the post processing is time consuming. As a consequence meeting minutes are published too late, incorrect or imperfect. And then, the protocols pile up on the desk - unread - and information get lost and assigned tasks are not done by the next meeting.

Our proposal represents the combination of existing speech recognition with word spotting techniques and an additional analysis, operation, and preparation for an automatic report generation and meeting workflow support, based on the above mentioned concepts. Therefore,the framework combines speech recognition and meeting management theories. Aim is to instantiate a lightweight, usable and intelligent tool for capturing, sharing and storage of information. This knowledge management tool has to handle externalized knowledge in a software development project and will confront speech as a fuzzy input.

In a second step it is contemplated, to provide a mobile solution of our system to support ad-hoc meetings.

3. Acquisition of Information from Audio Files

This section describes the process of data detection during a meeting and the realized concepts. The system is based on a keyword spotting approach where particular keywords are recognized during a spoken dialog. Therefore, our framework comprises a meeting grammar with defined keywords, which allow an improved speech recognition. The specialty is a dynamic extending grammar - a self-improving system - where the project context is integrated to enhance the speech recognition over the project.

3.1 Meeting Taxonomy

Based on Bischof [10] and Bruegge [11] a meeting taxonomy was defined and afterwards dedicated keywords for each meeting type determined.
First we identified meeting elements - smallest, recurring units of a meeting. These are for example ”status information”, where the status of current tasks, project progress, etc. is discussed, or a ”problem definition”, the announcement or definition of an issue. Additionally, ”decision making”, with the elements of IBIS (questions, ideas, arguments and a solution) is a common meeting element. Planning, controlling, discussions, brainstorming, or the scheduling of the next meeting are further units of a meeting.
Each meeting type consists of several of those meeting elements. A daily status meeting, for instance comprises ”status information”, possibly ”problem definitions”, if a team member face a problem, ”controlling” or the update of the project plan and estimations. Our whole meeting taxonomy tree is illustrated in figure 1.

The meeting grammar is build on the gathered meeting elements. For each smallest, recurring meeting unit a meeting grammar was defined. Thereon the keyword spotting component is based. In the following, exemplarily two grammars are presented.

```
    grammar statusChange;
/** If the developer takes a Backlog Item to start realization
* or it is tested/ finished, the status of this item changes.
* Currently there are three different states open, in progress,
* and completed.
* @example: change status of Action Item 2 to completed.
*/
<statusChange>= <updateCommand>„Status of“[<itemType>]
<BacklogItem>„to“<status>;
<updateCommand>= „update“| „change“| „add [new]“| „revise“| „adjust“| „modify“| „set“;
<itemType>= „Action Item“| „Backlog Item“| „Impedimen“| „Problem“| „Issue“| „Bug“;
<BacklogItem>= 1 | 2 | 2453 ; //current Backlog Item identifier
<status>= „open“| „in progress“| „completed“;
```

```
    grammar issue;
/** Add new issue to a problem list (e.g. during the daily
* status meeting).
* @example There is a new issue: How can we assure the
* connection to component x?. End of issue.
*/
<newIssue>= „new“<issue>: <issuedescription>. „End of issue.“;
<issue>= „issue“| „impediment“| „problem“| „blocker“;
<issuedescription>= text ;
```

3.2 Meeting Process

Following the process of data detection during a meeting is summarized and how the meeting grammar is used for keyword spotting. At the beginning of a meeting, a video camera or audio recorder is started. Then, continually a video/audio is produced which will be processed by the speech recognition engine. Afterwards the document will be search through for pre-defined keywords according to the grammar. If a keyword is found, the kind of keyword indicates the further process steps which will be carried out afterwards. We take for instance the daily status meeting. There, each team

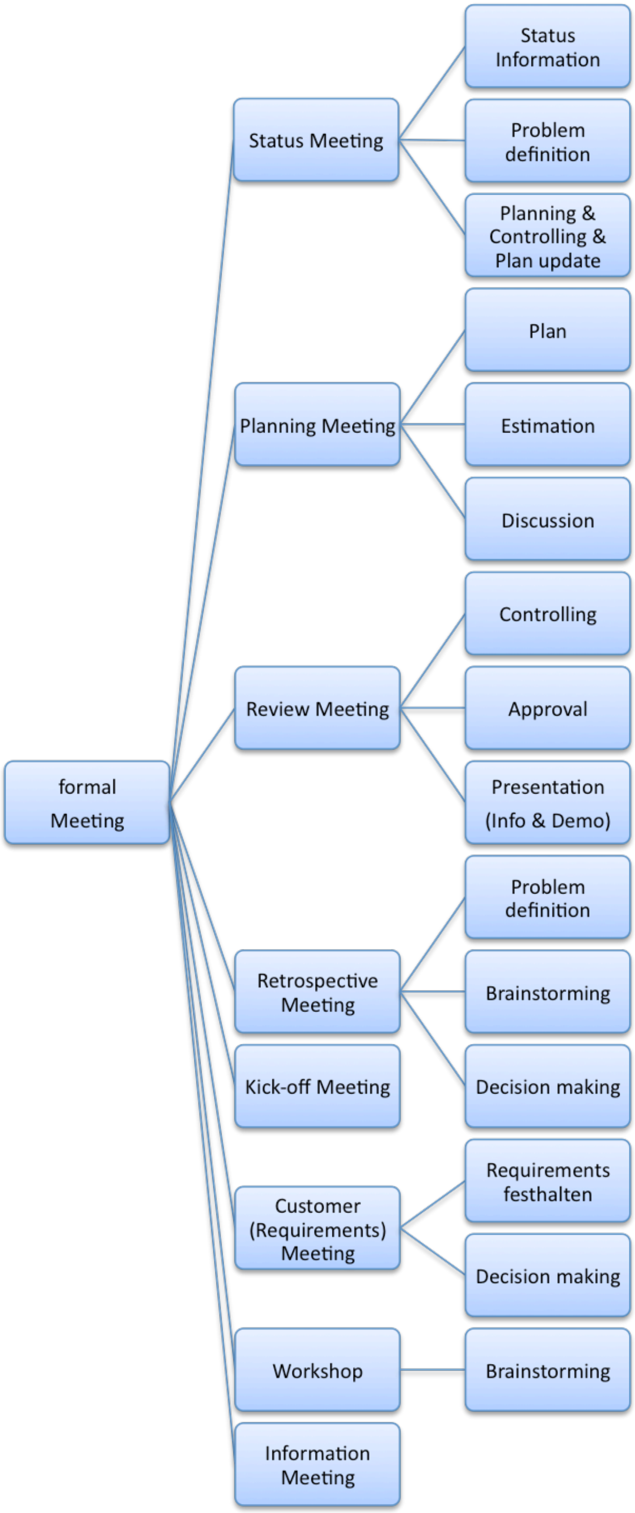

Figure 1. Meeting types and their elements

member presents its current work progress, whether a task corresponding to a requirement could be completed, is still in progress, or a new task was opened. If there are any impediments, which kept from doing the work, they will be reported, too. Moreover, new action items are defined to realize a requirement and assigned to a team member. There are special keywords for a daily status meeting to recognize status changes of a task, new action items, and new impediments, which are all summarized in the meeting grammar as proposed above.
The algorithm searches for these predefined keywords, which depend on the meeting type and processes the found information according to the keyword. So it is for instance possible to detect beside all action items current issues, status changes, as well as decisions, new software development requirements, and improvement ideas. Out of the meeting audio the information is accordingly filed and reports and protocols are automatically generated. These reports contain a summary of the meeting; but it is also possible to prepare a focused summary, e.g. all decisions and status changes which are connected to a dedicated requirement.

3.3 System Description

The system is decomposed into video processing and the meeting framework. The video processing unit accepts the video input, extracts the audio and forwards it to the speech recognition engine. Then, the output is send to the meeting framework, where a key word spotting - according to the predefined meeting grammar - is done. Afterwards, the found information is extracted and appropriately filed in the database, reports or meeting minutes are generated, and the information is forwarded to the appropriate project management tool. The whole system is illustrated in figure 2.

In regular intervals, the meeting grammar is extended by new project vocabulary. This is done by the framework by a search through project tools, like an design tool, for new project words. These are integrated in the meeting grammar, as it is most likely that the team will talk about those contents.

3.4 Interfaces

The specialty and uniqueness of our framework consists on the one hand in the predefined grammar to support an easy information extraction and on the other hand in a connection to other tools for project and document management.
Information, extracted in the meeting is either handed over to the protocol generator, or to the protocol generator and to a project management tool, e.g. a ToDo-List-Tool, if a new task was generated in the meeting. Consequently the discussed information of a meeting is accordingly filed and will never be lost.

This approach improves in addition the collaboration and the information transfer between participants of distributed

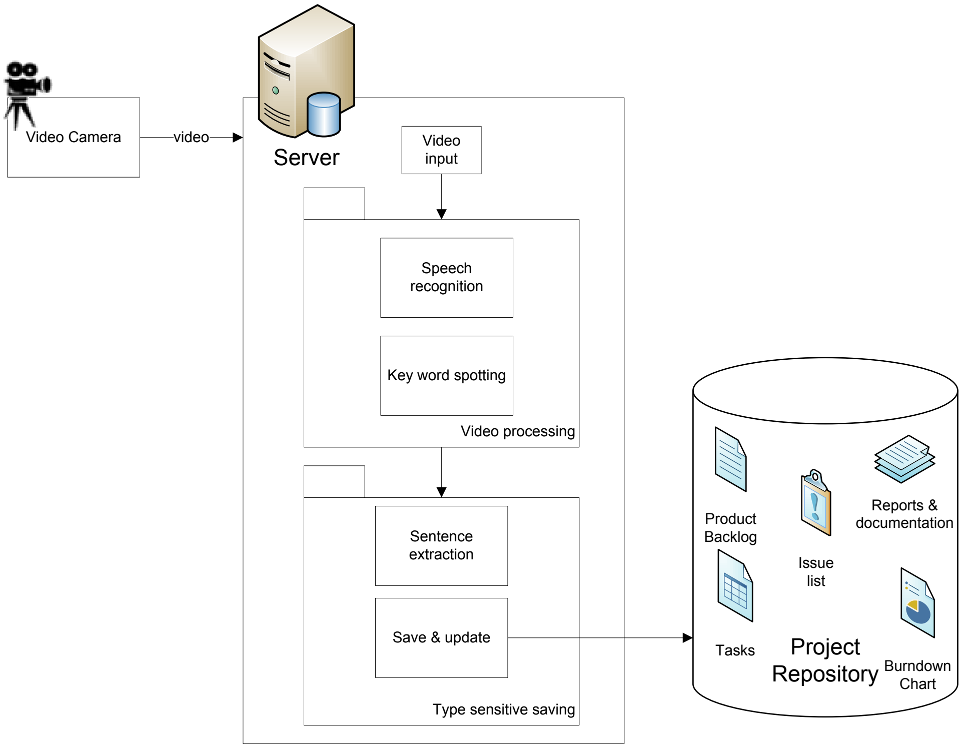

Figure 2. System decomposition

projects. Each team member can replay the decisions made in the meeting, even if he didn't attend the session. A special mechanism will notify him, if the requirement, he is working on, was mentioned in the meeting. Decisions, modifications, additional sub-items, etc. are automatically summarized in a protocol. The system provides spoken information that can be maintained and shared during the life of a requirement. So it is possible to go back and review all decisions regarding a requirement. This procedure is the first step to an automatic, companywide reporting mechanism and knowledge management approach.

4. Status Quo

Currently we are still in an iterative development and evaluation cycle. The main algorithms and functionalities are developed using the example of agile Scrum Meetings to show feasibility and evaluate the practicability of the idea.

Additionally an interface creation to the CASE tool unicase [12] is in progress. Unicase is a tool, integrating models from the different development activities, such as requirements, use cases, UML models, schedules, bug, and feature models into a unified model. This academically developed tool will allow us to create „real“action items for instance, detected in the meeting. They can be assigned to team members, attached with a due date and notifications are send automatically without any manual interaction. So the project team will be relieved by an additional work and rather can focus again on their actual work, creating software projects. Moreover, this tool provides in regular intervals project vocabulary, which is integrated in the speech recognition engine to enhance the word detection rate.

The present realization of our framework is arranged for commercial speech recognition. Currently we are evaluating several speech solutions like MacSpeech Dictate [13], the open source tool Sphinx [14], and Nuance‘s Dragon NaturallySpeaking [15].
Our requirements are in particular multiuser and untrained speech recognition, which are at the same time our biggest issues. We are searching for a pragmatic solution. To enhance the word detection rate several attempts are established, involving, instead of one microphone per room, a microphone per meeting participant and therewith one audio channel per user. Thus the speech recognition can be done per user with a user trained speech recognition, so we can improve the recognition correctness, as long the speech recognition research haven't progressed well enough.

The evaluation of the protocol generation and meeting workflow support framework is currently done in a student project at the Technische Universitaet Muenchen and is planned for real software development projects with industry cooperation partners like Siemens for fall 2009.

5. Conclusion and Outlook

This paper gave an overview of our framework to support an information extraction and retrieval out of meeting recordings.

First evaluations showed a consistently positive feedback. Our framework is a great assistance in software development projects by an advanced meeting management systems that supports a coherent workflow for information generation and retrieval. Our biggest issue is currently the speech recognition correctness as mentioned above.
Despite this, there are a lot of ideas to enhance this framework. As next steps an extension by a linking of the location in the audio and the information in a text document, for example is planned. Additionally, we are currently enlarging our meeting taxonomies (created for the classification of meeting types with associated keywords) by software development independent meetings. The idea is to support also other meeting types, especially the assistance and encouragement of ad-hoc meetings with a mobile solution of our system is currently focused and refined.

However, information comes not only from meetings. There are e-mails, forums, chats, and lots of other sources where information is exchanged in a communicative way. In a further step, these sources have to be integrated in the companywide knowledge and report mechanisms.

References

[1] Kunz, W.; Rittel, H., Issues As Elements of Information Systems. Working Paper No. 131, 1970

[2] Agile Manifesto, URL: http://agilemanifesto.org (access: 12.06.2009)

[3] Schwaber, K., Beedle, M., Agile Software Development with SCRUM. Prentice Hall, 2001

[4] Mountain Goat Software, URL: http://www.mountaingoat software.com/daily-scrum (access: 20.06.2009)

[5] DOLLI Project, URL: http://www1.in.tum.de/static/dolli/ (access: 12.06.2009)

[6] Bruegge, B., Reiss, M., Schiller, J., Agile Principles in Academic Education: A Case Study, itng, pp.1684-1686, 2009 Sixth International Conference on Information Technology: New Generations, 2009

[7] Doyle, M.; Straus, D., How to make meetings work. Jove Book, 1976

[8] Kayser, T. A., Mining Group Gold - How to Cash In On the Collaborative Brain Power of a Group. McGraw-Hill, 1995

[9] Sofotex, Agenda Organizer Deluxe, URL: http://www.sofotex.com/ (access: 24.06.2009)

[10] Bischof, A., Besprechungen: Effektiv und effizient. Haufe Verlag, 2007

[11] Bruegge, B.; Dutoit, A. H., Object-Oriented Software Engineering: Using UML, Patterns and Java. 2nd edition, Prentice Hall, 2003

[12] Unicase, URL: http://unicase.org/ (access: 24.06.2009)

[13] MacSpeech Dictate, URL: http://www.macspeech.com/ (access: 24.06.2009)

[14] The CMU Sphinx Group Open Source Speech Recognition Engine, URL: http://cmusphinx.sourceforge.net/html/ cmusphinx.php (access: 24.06.2009)

[15] Nuance, Dragon Naturally Speaking 10, URL: http://www.nuance.de/naturallyspeaking/ (access: 24.06.2009)