

Interrupted Coordinated Activities

Georg Apitz, Ben Bederson, Ben Shneiderman

Human-Computer Interaction Laboratory

University of Maryland, College Park, MD 20742

geapi, bederson, ben@cs.umd.edu

301-405-4639

ABSTRACT

The main goal of this paper is to describe insights found from an exploration of the field of coordinated activities. Many things that we do in our daily lives can be seen as coordinated activities, that is, they are several seemingly independent activities that are coordinated by a common goal. For example if one wants to send a letter, a stamp and envelope is needed on top of writing the letter. Buying stamp and envelope and writing the letter is all coordinated by the goal of sending a letter. Since these coordinated activities consist of several activities that are executed seemingly independent of each other, they are prone to interruptions. Thus, the challenge arises in facilitating complete coordinated activities, where the users are supported in maintaining the flow [3, 5] in the execution of coordinated activities. We discuss scenarios of coordinated activities and how templates can be introduced to support the user to achieve successful completion of their coordinated activities.

Author Keywords

Flow, interactions, coordination, simple interfaces.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Planning a wedding, auditing a company, planning a vacation, teaching a class or scheduling staff for a company seem to be activities that are very different. But, upon further inspection they show several things in common.

First, they are coordinated activities, that means they consist of several discrete activities which are executed to achieve a common goal. Second, they

have a time frame which restrains all discrete activities. This time frame is usually split up into several stages where different activities are prioritized. Third, they all involve people in differing numbers and in different roles. Lastly, all of them are influenced by constraints which can change the order of the discrete activities, make activities obsolete or require additional activities.

The problem is that users often forget the dependencies among these coordinated activities. Often the common goal of this coordinated activities is maintained implicitly and never manifested explicitly by means of making it part of every discrete activity that contributes to the coordinated activity.

With this implicit approach, changes in discrete activities which could potentially have an impact on other activities in the coordination are modified as single instances. Thus leading to a manual check if other activities within the coordinated activity are affected.

Since all these activities are restricted by a set of constraints and not necessarily can be completed in one continuous flow, the question is, how is it possible to support the user in achieving as much as possible under the given constraints and then to move on to the next activity. But, later on also getting the information that a previously attended activity now is in a state which allows it to be advanced more toward its goal.

Many of these coordinated activities that are encountered in the every day (working) life follow a similar schema and often times had similar coordinated activities in the past. The use of templates and macros can support the user in effectively planning these activities as well as ensuring that the execution of these plans is robust, stays in a complete and correct state and builds on a skeleton that provides the user with content that is reoccurring.

The scalability of such a system is one of the key aspects to consider. Providing a tool that supports an individual as well as collaboration and synchronization in a group is the logical goal. Tools like Microsoft's macro recorder or Apple's Automator

provide the user with some functionality to produce templates and macros that can simplify reoccurring activities. The problem is that they are focused on system features and have difficulties handling constraints.

PREVIOUS WORK

Malone and Crowston [12] provide an excellent survey of coordination and describe the different areas of research that are involved in it. They described how the different areas approach the problem of coordination differently and lay out what common ground there is that suggests an interdisciplinary approach to the problem. Their simple definition is: "Coordination is managing dependencies between activities". This leads to the conclusion that without dependencies there is nothing to coordinate. Malone and Crowston describe different strategies that are used to handle dependencies such as first come/first serve, budgets, managerial decision etc. Several online systems address the issue of managing coordinated activities and collaborative work from the perspective of a manager as well as a team member [2, 4, 9]. They allow the user to consider budgets, deadlines and view the status of the different projects. An example of a shared desktop application is Kerika [11], where users can share projects or parts of projects but it does not take into account dependencies, deadlines etc.

The topic of concentration has been looked at from different groups and has been approached from several areas. It is at least a combination of Sociology, Psychology and Computer Science. Concentration can be given when a user is in an environment where there are no external distractions/interruptions or if the user is in a state of concentration where distractions/interruptions become unimportant. This state is also described as "being in the flow". The idea of "the flow" as uninterrupted stream of productive work time in balance between anxiety and boredom was introduced by Csikszentmihalyi [5], Bederson [3] tries to build the bridge from this psychology book to its application in the area of interface design. He uses the three main criteria defined by Csikszentmihalyi to stay in the flow (skill, concentration, maintaining control) to show how they could be achieved in interface design. Bederson points out that the three stages in the learning process known as 1) cognitive, 2) associative and 3) autonomous can be mapped to the user types of 1) novice, 2) intermediate and 3) expert users. Furthermore, he argues that the aspects of staying-concentrated/avoiding-interruptions can be improved in interface design by minimizing drastic modal message boxes and dialogue boxes that abruptly and completely distract the user. He also pledges for more trust in the user in form of user-control, instead of automatically turning off or hiding

options the user should make these decisions. As a proof of concept Bederson briefly describes a note taking application that adheres to these principles.

Shneiderman and Bederson [15] describe the general process that is involved in maintaining concentration in order to complete tasks and what obstacles current interfaces show. They propose that slight changes to existing interfaces as well as new research can greatly improve usability and support users in achieving their goals. They argue that by supporting the user in maintaining concentration a better use experience as well as less frustration for the user can be achieved.

The types of interruptions and what is considered an interruption has been looked at by several researchers. Czerwinski et al. [6] report on a diary study of task switching and interruptions that classifies and describes interruptions and how knowledge workers handle these. Milewski [14] uses phone screening as an example and a study base to show how users actively handle interruptions and by screening phone calls make informed decisions to agree to an interruptions or to ignore it.

Wiberg and Whittaker [17] describe a prototype that uses a minimum amount of negotiation to handle interruptions. Instead of reacting immediately completely to an attention request the users negotiate a interruption time using a lightweight interface. With TaskTracer, Dragunov et al. [8, 10] describe a system that records the users interaction and activity and uses this information to make predictions about the next thing the user wants to do. They also use this information to adapt the interface so that for example in the Explorer favorites the ones relevant to the current task are shown first and thus the number of clicks to find or store something can be minimized. Matthews et al. [13] describe a system that uses semantic and change information to support the managing of multiple tasks and allow the user to stay focused on the right task at the right time.

Iqbal and Bailey [1, 10] describe different approaches to predict the cost of interruption. One focuses on the mental workload of the knowledge worker to predict good times for interruptions and also describes a framework how the mental workload can be used as a good estimator and how well this approach can be automated. Their starting point is to specify the cost of interruption (COI) and using their observation and modeling of the task structure try to minimize this cost. Iqbal and Bailey show how the characteristics of the task structures can be used to compute the best times for interruptions. These times are usually close to task or subtask boundaries and have a small cost in resuming the interrupted task.

Tverski and Kahneman [16] look at the aspect of making decisions in uncertain environments. (This can be potentially used when a user schedules a

due/completion date for a task but is too optimistic and a look at the other tasks in the same time span might be used to correct that date in either direction.)

DEFINITIONS:

activity: An goal related action done by a single person or several people together.

discrete activity: A single independent activity that does not contain any sub-activities.

coordinated activity: An arbitrary combination of discrete activities that have a common "goal". They can be part of each other, depend on each other or coexist coordinated by the common "goal".

meta activity: An arbitrary combination of coordinated activities that serve to achieve a common "goal". They can be part of each other, depend on each other or coexist coordinated by the common "goal".

template: A static approach of providing a skeleton of a plan for a coordinated activity.

macro: A dynamic automated mechanism that can create predefined plans for coordinated activities.

SCENARIOS

In the following we list several examples of coordinated activities that we consider. They are not in any particular order. More detailed descriptions of all of these scenarios can be found in the appendix.

- Organizing a trip
- Organizing a Conference
- Editing a journal
- Writing a paper
- Writing Code
- Planning a party/wedding
- Taking a vacation
- Selling a house
- Teaching a class
- Supervising an audit team
- Auditing a company
- Patient treatment process description

As mentioned before, most of these scenarios seem very different and not related at all but they are similar in their structure. This is the main point we are making, that based on this underlying structure we can devise a system that supports the execution of all of these scenarios and others as well. The approach is the following. Breaking complex task into easier ones and recombining them to achieve complex behavior can allow to understand dynamic relationships. Following our definitions we break any complex task into its simple parts following the idea of "divide and conquer". In other words we see a complex task as a

combination of simple tasks and look for a way to support users in composing these simple tasks to achieve complex results. Following we describe the main hindrances and possible solutions.

INTERRUPTIONS VS. COORDINATED ACTIVITIES

Interruptions Inherently through their nature of being a series of activities, coordinated activities provide ample space for interruption and distraction which can result in incomplete or incorrect execution of the coordinated activity. This stems from the fact, that distractions/interruptions are the main category of things that interfere with concentration. And, concentration is necessary to correctly execute coordinated activities. At first sight interruptions can be categorized into two groups.

System Interruptions As system interruptions we consider things that are initiated by the system without any action of the user to trigger or prepare the interruption, like the windows notification bubbles or a program reminding of a software update. We describe this interruptions as *solely system initiated*. On the other hand there are notifications that are issued by the system but the user has some influence on them. For example if a new email is received the email-client can notify about that event, but the user, at least on a sub consciousness level knows about it and is able to turn off or modify the way the interruption takes place. Similarly, chat messages that pop-up only happen when a user has a chat client running and a setting that allows the application to jump to the foreground if a new message is received. We can call this interruptions semi system initiated.

User Initiated Interruptions The user initiated interruptions or distractions are much more difficult to classify and very difficult to influence. They depend very much on the person and on the task they are performing. Certain cues that appear can trigger the memory of something that sank into the background and now is recalled and gets the users attention which shifts from the current task to the one the user just remembered. The interruptions themselves can be also viewed from a different angle, the task related one. If an interruption is triggered by an information about a project or task the user is currently working on it might not be counted as an interruption. While in the case of an unrelated interruption the user would potentially feel more interrupted.

Facilitate Concentration

Plans A general concept that we can apply is the concept of plans. Usually if we intend to do something we have a plan before the action is performed. For example if we want to print a document that plan could consist of four stages:

- 1) select document to print

- 2) select printer
- 3) issue print command
- 4) collect print-out

This resembles what we have described as a coordinated activity consisting of four activities that are governed by one common goal. If the system knows which activities are part of the coordinated activity the users is in the process of executing when he is interrupted it can help users to continue where they were interrupted. For example if the user gets a system message that overrides the print-dialogue, presenting an update for something which the user has to dismiss or act upon. The user might want an indicator that he was in the process of printing something and get the option if he wants to continue with it. To support this kind of interaction the system needs to know about the whole plan, which would need to be partially inferred (unless the user specified what he is doing) and the stage at which the user "left" this plan (the last activity which was executed). For this to happen we need to know:

- the plan as a series of activities (user specified or inferred?)
 - ⇒ correctness attributes (what are valid choices within this plan)
 - ⇒ completeness attributes (are there several steps that depend on each other)
- the stage (current activity) at which the user is in the plan
- a unique identifier that connects user, time and plan this could allow the system to default back to the correct plan and the position where the user left the plan.

Reminders This aspect ties tightly into the idea of plans. Once the system "knows" that the user was executing some plan but got interrupted, how is this information used. Is the user automatically reminded, can the user specify how to handle the reminders or would a hybrid solution be best?

Motivation for staying concentrated

A system that aims to support coordinated activities has to address the aspect of motivation. What can be done to motivate users to stay concentrated?

First, clearly identify and show the responsibility for each activity that contributes to the coordinated activity and also show the responsibility for the whole coordinated activity. When users interact with an activity they need a clear idea in which relation they are to the activity in terms of responsibility. If a user knows he is responsible for an activity and that others depend on that activity, he is more inclined to execute the activity. Second, using rewards to motivate users can prove beneficial. Similar as in the case of interruptions [7] this is likely to only be successful if

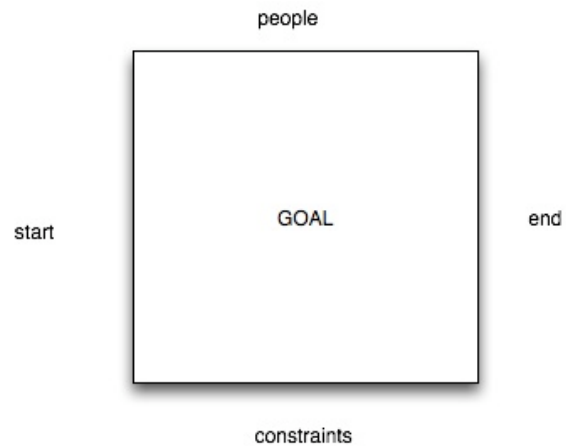


Figure 1. In this illustration of a coordinated activity there are five aspects visualized. The center of the coordinated activity is the goal surrounded by people, constraints and start and end date. This visualizes nicely the things that drive the coordinated activity and as long as it stays within these boundaries it is considered correct and complete.

the reward system works in both directions. Sometimes a user contributes and sometimes he benefits in a balanced manner. Third, the issue of trust which is tightly coupled with the previous two aspects needs to be transparent to the user. The user needs to know if he can trust the person in charge for a specific activity to deliver (on-time and complete). This could be achieved by a combination of rating by the user(s) and an inference of the system looking at prior activities the user to be rated was responsible for. Also, it has to be obvious how communication patterns are executed. If a user is responsible for an activity then he must also know if he has to propagate the information about the completion of the activity or if the user waiting on the activity does the checking or if the system handles that completely.

The main challenge here is to leverage these aspects so that the motivation for staying concentrated stays at a high level. For example if the responsibility for something can be shared, everybody involved has to worry less and can focus more on the aspects he has to deal with.

USERS AND SYSTEM DESCRIPTION

Potential users

The potential users of such a system are another challenge. Several of the scenarios we described tend to be orchestrated by one-time or sporadic users. In general people get married once, organize only very few conferences and have vacations once a year. Nevertheless the coordinated activities are always very similar and it would be beneficial to reuse gained experience and strategies.

On the other hand, auditing a company, writing source code, writing papers, reviewing papers, professionally planning weddings are coordinated activities that are done frequently by experts in the respective field. And the same is true here, reusing gained experience and strategies can be of great benefit.

That means a system that aims to address all of our mentioned scenarios has to provide a good interface for both user groups. Maybe a layered interface approach is helpful here.

Design Goals

The system we envision has two main aspects. First, it should be very lightweight and easy to use and second, it should support users to plan and execute coordinated activities with the common goal in mind.

Platform

Another question that needs to be answered is the way in which the system is realized. It could be web based or a desktop application. The earlier would allow users to access it everywhere, which includes office, home and mobile devices. But it also put some technical limits to the interface. Making it a desktop application bears the problem of synchronization but gives all the advantages of the desktop world. This would mean that the application could be used offline and users could use it in any preferred setting.

Template system

Using a template system in this context can go in different directions. One way is to automatically provide or let the user define templates of content. The other kind of template would be the coordinated activity template. If we imagine an interface that can just generally handle coordinated activities that follow the definitions from the beginning then templates could be used to load the specific interface for that specific coordinated activity. This could include adding more buttons, controls etc. to the interface or removing some, enabling adapted interactions, report/export functions, turn on/off communication, synchronization with co-workers etc.

A third option is to use templates a propagation method, where the organizer of a conference for example can put a template for the conference on the announcing webpage. Interested users can then download the template into their calendar system and use it as scaffolding to plan their individual visit to the conference. This has the benefit that individuals do not have to worry about deadlines etc. since paper, registrations, submission deadlines etc. are the same for thousands of users. One could even imagine that the user can specify in which function he is related to the conference (attending (A), presenting(P), reviewing (R), A+P, A+R, A+P+R etc.) and based on

that specific information is put into the template the user requests.

Integrate the user

Giving users the ability to shape the system to their best use can be intimidating or rewarding. Novice users can easily be overwhelmed with too much functionality and too many options. Expert users on the other hand prefer highly customizable, efficient interfaces. Providing a system that addresses both of these user groups and allows the transition between them needs to be highly flexible. Starting off with the minimum functionality and providing more and more with the increasing comfort/knowledge level of the user can help elevating the problem of different users.

For example when the user starts using the system it could present several questions and based on the users answers the interface can be adapted. Also, the user could have the choice between different settings that could be based on "model" users and then adapt these initial suggestions to their needs. In some minimalist cases and also setting like meetings or in airport terminals, a very simple close to command line interface could be very powerful. This could allow the user to make use of the system even in very specific environments, that either require the majority of the users concentration or are very disturbing per se so that there is not a lot of time or room for interacting with the system. This short-yet-powerful notion is very crucial if the system aims for a wide use. These are all just initial suggestions and need to be verified and most likely extended or change in collaboration with actual users in real life settings and user studies.

THE PWC-CAR EXPERIENCE

Interactions with a very specific user group (namely auditors) showed several valuable points that tie into the previous observations and suggestions. Since these groups do not have any dedicated software tools to support the planning of coordinated activities they rely heavily on "old fashioned" techniques. These are especially whiteboards, sticky notes, notes in general and emails. They come together with severe problems. The biggest one being synchronization. If information is written on a whiteboard at location X, it is only accessible at that location and neither reading nor writing to it from a different location is possible. Furthermore, notes and emails that are exchanged between individuals are only available in their individual system. This may result in duplicated information, overlooked assignments, missed deadlines, short: frustration (in a economical and social sense). This system is very personalized and most of the synchronization and consistency checking has to happen in one (manager) or several persons head.

Another aspect to be considered is documentation, whiteboards, notes and emails are difficult to document and archive. They are accessible to individual people but not to a whole group per se.

Looking at the kinds of coordinated activities that are executed in this specific area we found that many of these activities are repeated over and over again. Timelines and work plans are set up several times a year following very similar patterns. But with the approach so far, each time a new plan is drawn from scratch on the whiteboard, countless emails are sent out again and some deadlines are overseen or missed.

A question to the users which feature they would like to have in a potential system was commonly answered with "reusability of repeating steps and easy access to all the information from everywhere".

Many of the managers had looked at available software solutions but found them to be too cumbersome to learn and use and to contain much more functionality than they wanted or not the desired functionality. Also, company policy plays an important role in decisions about which software/technique to use in a given setting.

Another challenge in a production environment, like the auditing field, is the integration of a new system into a running procedure. Providing a new system is nothing that can be introduced gradually. The system needs to be developed up to a point where everybody agrees on its readiness for production and then it needs to be used in practice. This increases the fear of such a change and the number of demands that are posed to the system before it is even approved for use in the "field".

FUTURE WORK

Conference Planning Tool One next step could be to implement a system that is geared toward one user group, like conference planners, and use this user base to initially test the system. Conference planning seems to be a good starting point since it is a common problem in Academia and users should be easy to find. Also, it has a nice applicability by addressing the problem of different user types really well. From the one-time (submitter, reviewer) user to the regular (editors, committee members) user all user groups are involved and their feedback is likely to be very valuable. Then, when several insights have been collected and several iterations in collaboration with users are done an extension or generalization can be looked at again.

Evaluate/Combine/Extend Current Systems Another approach could be to use initial user studies on existing systems such as *InTheKnow*[9] and *BaseCamp* [2] to find shortcomings and identify the details that users would like to find in them and define features that should be added or can potentially

be found by combining different systems. The in the next step these modifications could be studied again and ultimately show if a new or combined solution would address the problem better.

Other existing techniques The problem of managing coordinated activities is not new, so a third path to take is to look at traditional ways of coping with the problem. Any craft with complex workflows deals with a certain set of coordinated activities. And, presumably each of them has developed ways of ensure the correct execution of the activities that constitute the coordinated activity. What can we learn there, how can modern systems benefit from these insights and make use of the developed techniques?

CONCLUSION

In this paper we described the problem of coordinated activities and similarities between different coordinated activities. Several "managing" tasks can be seen as coordinated activities and they share common attributes that can be used to build an interface that can support users with their coordinated activities. We described several scenarios and the correlation in structure between them. We also reported on some insights found from working with subjects in the field of auditing and their current approach of handling coordinated activities. Furthermore, we show some future directions to evolve the topic to.

REFERENCES

1. Bailey, B.P., and Iqbal, S.T. "MeWS-IT: A Mental Workload based System for Interruption Timing" In Proceedings of *UIST 2005, Doctoral Symposium* Seattle: 2005.
2. *Basecamp* <<http://www.basecamp.com/>>.
3. Bederson, Benjamin. "Interfaces for Staying in the Flow" In Proceedings of *TR 2003-37*. University of Maryland, Human-Computer Laboratory, 2003.
4. *Chandler* Open Source Applications Foundation. <<http://chandler.osafoundation.org>>.
5. Csikszentmihalyi, Mihaly. *Flow: The Psychology of Optimal Experience*. HarperCollins, 1991.
6. Czerwinski, Mary, Eric Horvitz, and Susan Wilhite. "A diary study of task switching and interruptions" In Proceedings of *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems* Vienna, Austria: ACM Press New York, NY, USA, 2004. 175–182.

7. Dabbish, Laura, and Robert E. Kraut. "Controlling interruptions: awareness displays and social motivation for coordination" In Proceedings of *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work* Chicago, Illinois, USA: ACM Press New York, NY, USA, 2004. 182–191.
8. Dragunov, Anton, N., Thomas Dietterich, G., Kevin Johnsrude, Matthew McLaughlin, Lida Li, and Jonathan L. Herlocker. "TaskTracer: a desktop environment to support multi-tasking knowledge workers" In Proceedings of *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces* San Diego, California, USA: ACM Press New York, NY, USA, 2005. 75-82.
9. *In The Know!* <<http://www.intheknow.info/>>.
10. Iqbal, S.T., and B.P. Bailey. "Leveraging Characteristics of Task Structure to Predict the Cost of Interruption" In Proceedings of *CHI 2006* 2006.
11. *Kerika* < <http://www.kerika.com/>>.
12. Malone, Thomas W., and Crowston, Kevin. "The interdisciplinary study of coordination." *ACM Comput. Surv.* 26 (1994): 87–119.
13. Matthews, Tara, Mary Czerwinski, and et al. "Clipping Lists and Change Borders: Improving Multitasking Efficiency with Peripheral Information Design" In Proceedings of *CHI* 2006.
14. Milewski, Allen E. "Interruption Management and Telephone Call Screening." *International Journal of Human-Computer Interaction.* 20 (2006): 19-33.
15. Shneiderman, B., and B.B Bederson. "Maintaining Concentration to Achieve Task Completion" In Proceedings of *DUX* 2005.
16. Tversky, A., and D. Kahneman. *Judgment under uncertainty: heuristics and biases.* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1990.
17. Wiberg, Mikael, and Whittaker, Steve. "Managing availability: Supporting lightweight negotiations to handle interruptions." *ACM Trans. Comput.-Hum. Interact.* 12 (2005): 356–387.

APPENDIX

Scenarios

1) Organizing a trip

Scenario of planning a trip to a conference in Switzerland for 15.-17. October 2006

Detailed description here, abstracted below:

The first step was the decision to fly to Switzerland in October this started the coordinated activity of planning a trip to Switzerland.

first (also planning) stage:

In this coordinated activity the 1)first activity is to find and book a flight. Outgoing flight on the 14th of October (time difference) after a 3 o'clock meeting and the returning flight to be back to teach at the 19th of October. These are the two main constraints on the coordinated activity of the Switzerland trip. 2) After the flights are booked the next activity is to find and book a hotel (this could also be done in conjunction with the flight booking, depending on the approach), but here the recommendations of the conference are used and one of their hotels is booked. 2a) Arrange transportation to airport and from airport to hotel. A cab is planned for the trip to the airport and the conference offers a shuttle service to the conference hotel.

3) After that the next activity is to look at the conference schedule and mark 7 potentially interesting papers to reserve time for their presentations and also to reserve time for social and general events. 4) The next activity is to schedule meetings with people of interest. Two emails with requests are available as well as 3 other meetings that are stored in the users head. After looking at the reserved spaces for the paper presentations and social activities that are already on the conference schedule 5 spots are found and allocated for the meetings.

second stage (the execution of the coordinated activity):

0) Pack.

1) Take the cab to the airport. 2) Fly to Switzerland. 3) Take shuttle to the hotel. 4) Attend the keynote. 5) Attend first paper presentation of interest.

During that presentation a colleague talks to you who has not seen you in a long time and you arrange to meet over lunch with him. 6) Attend second paper presentation of interest. 7) Meet colleague over lunch. 8) Attend third paper presentation of interest. 9) Meeting 1. 10) Meeting 2. 11) You get a call from a colleague who cannot make it to the meeting. 12) Meeting 4. 13) Attend fourth paper presentation of interest. 14) Attend fifth paper presentation of interest. During the presentation you talk to the colleague you were supposed to meet later but he has to move the meeting. You leave the presentation together and skip the next one of interest and meet instead. 15) Attend seventh paper presentation of interest. 16) Attend town hall meeting. 17) Take shuttle to the airport. 18) Fly back 19) Take cab home. 20) Unpack.

third stage (after the event)

- 1) File for reimbursement.
- 2) Send emails you promised during the conference.
- 3) Inform your colleagues about the conference.

What is involved in a planning a trip?

goal: traveling to a specific destination (with a specific intention)

stages:

- before the activity
 - select destination (if not predefined)
 - arrange date (if not predefined)
 - arrange accommodation
 - arrange travel (car/flight)
- during the activity
 - attend trip activities (meetings, presentations, theater etc.)
- after the activity
 - report on results
 - send during trip requested information
 - file reimbursement statement

people involved:

- how many
 - one to many
- in which roles
 - organizer or participant or both

organizer:

- organizes for himself
- somebody else

granularity:

- spans several weeks (day as minimal granularity)

constraints:

- sickness
- weather
- time line
- current agenda

expectations:

- serve the goal of the trip

2) Organizing a Conference

goal: create an important event for a specific research community

From my thinking about it so far there are three kinds of people who

are mainly involved. One being the conference chair, one being the committee chairs and one being the committee members.

The committee chair is similar to a director of a movie, he has to find his crew namely the committee chairs and the committees and then make sure that the organizational process happens smoothly. This is similar to a partner at PwC who is in charge of several engagement teams and has to make sure that all of them run smoothly.

My definition is as follows. The chair of the conferences is the general manager in charge who appoints and manages the committees on a high level and makes sure that the overall process of organizing the conference runs smoothly. The committee chairs are committee members with "special" tasks, namely managing the committee they are heading and making sure that the specific work in that committee is done correctly. The committee members are in a similar situation but they "only" make sure that their part in the committee is done and that it fits into the whole process.

What we see here, are the typical levels of these coordinated activities. At the lowest level most of the "hard" labor is done, like finding a caterer and negotiating prices for a specific number of meals. One level higher the committee chair has to make sure that this activity is executed and check on the result at a specific time, but he is not interested in all the details.

Another level higher the conference chair mainly wants the information from the committee chairs if things go smoothly in the committees. The higher the "level" the more coordination and managing work as opposed to "real" work has to be done.

Describing it in the coordinated activity view would be as follows.

Organizing the whole conference is the meta activity, which contains the meta activities of program, student volunteers, posters, demos, proceedings & DVDs, public relations, registration etc.

Each of these meta activities contains coordinated activities that serve together the goal of the meta activity.

The following scenario outline is for the student volunteer meta activity.

- before the activity
 - announce that the conference is looking for SVs
 - synchronize with the conference chair about activities that are done by the SVs
 - set a deadline for applications
 - collect and select applications
 - contact selected (not selected) students with general information
 - set up a wiki for the SVs
 - maintain communication with SVs about acceptance and details
 - ask SVs about schedule preferences
 - set up a schedule for the SVs
 - bagging, badge producing/assembling
- during the activity
 - execute requested activities:
 - registration

- AV support
- badge checking
- guide/help services
- check the schedule and supplies for the SVs
- adapt schedule if necessary or additional activities arise
- organize SV party
- find new SV chairs

- after the activity
 - thanks SVs
 - coordinate with next committee
 - write up experiences

people involved:

- how many
 - SV chairs and SVs
- in which roles
 - organizer, "worker"

organizer:

- chair

granularity:

- spans several months (day as minimal granularity)

constraints:

- time (conference date)
- schedule preferences of the SVs
- limited amount of SVs
- certain SVs might not be able to do certain activities

expectations:

- to provide a smooth experience for the attendees

3) Editing a journal

Goal: acquire high quality journal content

The process of editing a journal follows the concept of meta and coordinated activities nicely as well as that it has different levels with different activities. On the top level the editor(s) handles the meta activity of editing a journal, which involves the meta activities of submission, review, acceptance, rejection and dealing with the publisher.

Each of these meta activities again contains several coordinated activities that serve the goal of the meta activity. The editor has to make sure that the applying meta activities are executed correct and complete.

Following the scenario outlines for the different meta activities:

submission:

before

- advertisement
- call for papers
- setting up a submit service

during

- upload/sent paper
- correspondence with the author(s)

after

- information about successful/unsuccessful submission

review

before

- find reviewers
- find out how many papers a reviewer is able to review
- distribute papers to review
- announce deadline for the review

during

- do the review
- check status of the review
- send reminder(s)

after

- collect the reviews
- create meta review
- thanks reviewers

acceptance

before

- review

during

- inform author(s)
- process changes
- review

after

- send final acceptance note
- send to publisher
- send copy of journal?

rejection

before

- review

during

- send note to author(s)

after

dealing with the publisher

before

- negotiate size of publication and price
- negotiate print deadlines

during

- send papers
- review results

after

people involved:

- editor, reviewers, authors
- in which roles
 - manager, consumer, producer

organizer:

- editor(s)

granularity:

- spans several months (day as minimal granularity)

constraints:

- topic
- deadlines
- reviewer capacity
- number of papers per edition

expectations:

- produce a good journal

4) Writing a paper

goal: write a paper that gets accepted at a good venue

The paper writing is a good example of one meta activity (writing a paper) which entails several coordinated activities as outlined below.

stages:

before the activity

- identify an interesting topic
- recruit coauthors

during the activity

- literature review
- do a study/produce results on which the paper is based on
- produce a draft of the paper
- evaluate results
- evolve the paper
- get internal reviews

after the activity

- submit the paper
- rebuttal
- update and finalize
- present at conference

people involved:

- authors, reviewers
- in which roles
 - producer, consumer

organizer:

- author(s)

granularity:

- spans several weeks (day as minimal granularity)

constraints:

- topic (has it been done before)
- deadline
- subject availability
- other commitments of authors

expectations:

- produce an accepted paper

5) Writing Code

goal: write code that solves a given problem

stages:

before the activity

- identify the problem to solve
- structure the problem
- identify main parts of the code
- select a language

during the activity

- write code
- write documentation
- test code
- have a finalized application

after the activity

- maintain code
- bug fixes
- produce new versions

people involved:

- authors, testers
- in which roles
 - producer, consumer

organizer:

- author(s), source code manager

granularity:

- spans several weeks (day as minimal granularity)

constraints:

- technical resources (hardware, licenses)
- deadline
- available libraries
- other commitments of authors

expectations:

- produce good and working source code

6) Planning a party/wedding

goal: celebrate an event

stages:

before the activity

- arrange date
- arrange location
- arrange catering
- arrange accommodation
- set up guest list
- send invitations
- confirm catering
- check invitation list

during the activity

- guide through the event

after the activity

- send thank-you notes
- clean-up

people involved:

- how many
 - many
- in which roles
 - host, attendees

organizer:

- bride/groom, parents, host

granularity:

- spans several weeks (day as minimal granularity)

constraints:

- holidays
- round birthdays of close relatives

expectations:

- a happy event for all involved

7) Taking a vacation

goal: taking a vacation

stages:

before the activity

- select destination (if not predefined)
- arrange date
- arrange accommodation
- arrange travel (car/flight)

- stop mail
- have somebody water the plants

during the activity

- enjoy the trip
- keep a diary
- send postcards

after the activity

- resume mail
- check with person who checked the house if everything is alright
- get pictures developed/printed

people involved:

- how many
 - one to many
- in which roles
 - organizer or participant or both

organizer:

- organizes for himself
- somebody else

granularity:

- spans several weeks (day as minimal granularity)

constraints:

- sickness
- weather
- vacation availability

expectations:

- relax

8) Selling a house

goal: sell house

stages:

before the activity

- contact realtor
- set expected price
- take pictures
- write description
- move out
- clean the house
- make counter offer
- redirect utilities to new place or terminate

- during the activity

- accept offer
- check if money transfer worked

- after the activity

- pay realtor
- manage final bills

people involved:

- seller, buyer, realtor

organizer:

- realtor

granularity:

- spans several weeks (day as minimal granularity)

constraints:

- min price
- deadline

expectations:

- sell house to a specific minimum price

9) Teaching a class

goal: teach a good class

stages:

before the activity

- prepare syllabus
- prepare lessons
- prepare projects/assignments
- prepare website
- set up mailing list

during the activity

- assign homework/projects
- collect/grade homeworks/projects
- answer student queries
- teach
- hold office hours
- schedule substitutions
- fine tune lesson preparation

after the activity

- submit final grades
- archive lecture material
- deal with re-grades

people involved:

- professor, students

organizer:

- professor

granularity:

- spans many weeks (day as minimal granularity)

constraints:

- university schedule
- other commitments of professor
- holidays

expectations:

- teach an interesting and insightful class

10) Supervising an audit team

goal: efficiently manage one engagement team (potentially in different locations)

The general problem with the engagement team management right now is that all the information is either on a whiteboard or in the head of the manager. It starts with the managing of people, not in the sense of scheduling (the staff works pretty independent) but in the sense who does which steps of the audit procedure by when.

This happens repetitively (every quarter + annually + for 404 even more often), does not change drastically for a specific client and is so far done on a white board and in the managers head. This means the manager doesn't have the information if not on-site, this becomes especially important for big engagements (big client, several locations).

They need an easy way to set up the coordinated activity of the engagement and to keep track of who does what by when (which steps), who reviews what.

Since the preferences of managers in terms of reviews are pretty static (exception when new staff comes in the team) they could specify preferences what they want to review personally and what they trust their senior to review

and that information could be used automatically to make assignments.

Another thing that is important in assigning steps to an individual is the knowledge about that persons skills (previous engagements, sections they worked on) as well as contact information in case information is needed in "off" hours.

This fits into the idea of coordinated activities with a specific twist of the underlying information about preferences and skills. For repetitive coordinated activities like the assignment of steps to people templates could be used. The engagement would be the meta activity here and the different steps and procedures be coordinated activities.

stages:

before

- set up/update preferences about who has to review what

during the activity

- set up a schedule with the team
 - who does which step
 - who reviews what (partially automated)
- answer requests from staff
- check status of specific steps
- check status of specific team members
- deal with requests (include notifications, if not returned by certain date)
- update the schedule

- after the activity

- update templates and review preferences

people involved:

- partner, manager, staff

organizer:

- manager

granularity:

- spans many weeks (day as minimal granularity)

constraints:

- deadline

expectations:

- have a successful and valid audit

GOAL:

A collaborative web based tool that supports planning and monitoring of the audit process for an engagement team.

The tool should mitigate the problem of coordinating the specific challenges in managing an engagement team.

It should support the engagement team in brainstorming and planning sessions that are traditionally supported by whiteboards. This process should be supported by Smartboards which can be used like the traditional whiteboards but offer a history and documentation functionality as well as providing the content in digital form. The tool should also make background information available that is traditionally kept in people's heads or only accessible via other resources like HR. This would be things like preferences of the manager for who reviews what and experience profiles of the individuals on the audit team.

In a first step an initial prototype should be revised that shows the basic functionality of the web application and typical interactions on the whiteboard could be replicated using smartboards.

The general goal of the tool is to support the management of the audit process and to avoid problems due to missing or unaccessible information in the managing process.

features:

- plan who works on which section and completion date
- who reviews what
- set trust levels for reviews (i.e. o.k. if only done by senior etc.)
- ability to display a schedule overview similar to the current white board approach
- store plans as templates since they are repeating every Quarter
- give an overview which sections are completed and who reviewed them
- give an overview by person, who is doing what and what is the status
- provide information about each team member (skills, prior engagements, sections they covered [area: software /revenue and other assets])
- keep track of requests (i.e. documents from client) and notify if not returned by deadline

11) Auditing a company (minimum predictability)

goal: successful audit of a company

stages:

before the activity

- identify risks
- create an audit plan
- select the team(s)
- set up a schedule for the team(s)
- update the schedule
- update audit plan

during the activity

- answer requests from team(s)
- check schedule
- check documentation
- deal with client requests

after the activity

- archive audit

people involved:

- partner, manager, staff

organizer:

- manager

granularity:

- spans many weeks (day as minimal granularity)

constraints:

- deadline

expectations:

- have a successful and valid audit

12) Patient treatment process description

Patient Smith is delivered to the ER and is checked in by Doctor Foo. After an initial evaluation of the patient Dr. Foo uses PlanBar to set up the coordinated activity of the treatment of the patient. Since Dr. Foo diagnosed Smith with X she can use the template for a standard treatment of X and modify specific entries for this patient. The template suggests a lab test activity first and depending on the outcome either reevaluation or medication and treatment activities followed by a revaluation activity after one week. Dr Foo goes ahead and sets the first activity for this patient

as a lab test to confirm or reject the diagnosis. She sets a deadline for the test results in 24 hours and indicates to the system to notify her on her pager if the test fails.

For the case that the lab test confirms the diagnosis Dr. Foo modifies the medication activity.

She advises the administration of medication z 3 times a day and for the medication d once a day.

Furthermore, she sets the treatment activity to be a reflex training scheduled for once a week.

The revaluation activity is set for 5 days after the initial check in.

goal: treatment of a patient

stages:

before the activity

- evaluate patient
- create an treatment plan
 - lab tests
 - medications
 - treatments
 - radiology studies
- consultations with other doctors

during the activity

- request lab tests
- administer medication
- apply treatment

after the activity

- send patient home
- schedule time for check up

people involved:

- doctor(s), nurses, patient

organizer:

- doctor

granularity:

- spans up to many weeks (day as minimal granularity)

constraints:

- allergies
- risks

expectations:

- have an suitable treatment for a patient