Metaprojects and partial tasks

Elements complementing the classical project structure and supporting efficient group work management

Andrzej Krawczyk

Synopsis

The article discusses the limitations of the classical methodology of group work management and presents our proposal of complementing it with a special type of relation between the project and process (cases and partial tasks) as well as associating all the operations of an organization with a superior rank structure (called a metaproject) which will enable access to knowledge on the contexts of these operations.

Introduction

The fundamental role of information in the operations of any organization cannot be overstated. Information has now become the highest-priced commodity and the access to it and its adequate use are now critical for achieving success. Nowadays, the technological advance has overcome the practical obstacles and enabled collecting and storing virtually unlimited amounts of data. However, still unresolved is the issue of appropriate organizing of the information, i.e. turning it into real KNOWLEDGE.

Managing the work of a group of people with the help of the classical planning tools is efficient only up to a point. For it provides general information on individual tasks carried out by the employees without bringing any details on their performance. Another major limitation stems from the fact that the traditionally applied project structure is oriented towards achieving goals with no respect to broader context of project operations.

The article lays out our proposal for complementing the methodology of group work management that will help to overcome these limitations. The proposal is grounded on a meticulous analysis of the specificity of the issue and on the experience gathered from our everyday work with the HumanWork 1.x system; our conclusions have then been practically implemented in HumanWork 2.0.

Types of information in group work

The most fundamental classification of information involving the work of a group of people differentiates between information on actions and their results..

Document - action result description

The results of individual people's actions can have various nature: a manufactured box of nails, cleaned office, cured patient, money transfer made or a written article. Therefore, it is impossible to define a general and complete set of attributes describing all the potential action results. The only sensible solution for a software system is to reflect the actions' results as DOCUMENTS with their individual properties and presentation adequate for the type of information they carry.

The support that a software system can, and should, offer in this respect involves wide range of tools for grouping, categorizing and linking documents. The users should be able to create folders of varying structures, both public and private, and to add a document to any selected number of the folders. Additionally, they should be enabled to link documents with each other and to connect documents to other vital data items in the system.

Task - action description

Unlike action results, impossible to define within one universal form, human actions as such can be categorized in this way, irrespective of the specific nature of the actions involved.

The elementary form of action description is a task. Irrespective of their nature, all tasks have numerous universal features resulting from the basic fact that they are all positioned in time, they all require resources and that their results can be qualified in general terms.

Collecting information on tasks is available in many software systems. The basic set of tools can be found in applications of the Personal Information Manager class, presenting the list of tasks to be performed on the given day or in any other selected time range. In programs like these, often the only item of information characterizing a task is a one-line text and, if available, the task performance dead-line.

Most advanced in this field are project management systems. The information on tasks they can process is very detailed. They enable defining the range and allocation of time, set up time constraints, assign costs or allocate required permanent and consumable resources. The project management theory applied in them is reflected in the project structure and links between tasks.

Project management theory vs. reality

In spite of all the developed methodology, applying the rules of the management theory in the context of the life of an entire organization poses numerous problems. These problems do not result from technical limitations or imperfections in the systems - on the contrary: they are products of the very principles on which the whole methodology is based.

Project management systems with even the most sophisticated tools are all based on the premises that each task at every stage of project execution is planned and monitored by somebody. Observing this principle strictly for every single action taken within the planned operations places excessive burdens on the medium-level management of the organization and causes significant delays in the execution of every individual task and - as a result - delays the whole project. In consequence, only some projects are run in line with the principles of project management methodology, and even these only to some limited extent.

The reasons behind these significant restrictions on practical use of the classical management methodology are simple. As long as the number of executed actions and their level of detail does not overburden an individual - the method works flawlessly. Problems begin to arise when the individual takes on managing too many factors of the operation. To paraphrase the principle of indetermination: the more detailed the management of an operation is, the slower and more costly it is executed. Along with the increase in the level of detail fed to the system comes a better control over the process but the time devoted to defining and monitoring the information sky-rockets until it actually becomes longer and more costly than the project itself.

Arriving at the 'healthy proportions' should come down to a proper generalization of the project-related information fed to the system.

Solving the problem of the practical limit to the detail in project mangement was one of the chief targets for HumanWork. This proved impossible without expanding the classical project structure. In order to introduce in a proper way the importance of the problem and the significance of our solutions, let us first discuss the first stage in the development of the system when it still kept the classical project management approach.

Describing process through tasks

During the first stage of system development we solved the problem of time-consuming data definitions for the project course thanks to the application of process management and document-flow solutions.

The idea came from the observation that at certain level of detail the task structures show significant similarities and fall into structures which can be organized with the help of procedure diagrams. This results from the fact that, although every project has a very individual nature, at some level of generalization we actually come across operational desciptions whose sequences can be best expressed in the form of procedures.

In this way, we distinguished in the planning process the planning of individual undertakings from the modeling of repetitive processes. We also implemented the ability to create links between the project and a ready-made procedure by defining the procedure as a case within the project.

This enabled us to reduce the planning of the given project to the very general level and to use at its more detailed levels certain patterns, i.e. procedures, which model the exact course of proceedings. The use of a procedure in the project (i.e. running a case) is a command for the system to generate a set of tasks previously modeled in the procedure.

Additionally, to enable updating the project course, all the data (planned individually and automatically generated from a procedure) are transferred to the people responsible for tasks execution. These individual workers (with the help of a tool which effectively combines an organizer and a messenger system), while performing their allocated tasks, automatically supply information on the work progress on their tasks, which in turn contributes to the information on the progress of the entire project.

The full effectiveness of work with our system has been achieved through implementing the option of adding to any task the documents that are needed for the task performance. The documents can be added 'manually' or automatically, based on the document-flow rules defined in the project and procedures.

Tasks volume problems

The system features mentioned above enabled detailed monitoring of the executed projects. Unfortunately, this also brought about a large increase in their volume, which - despite the use of advanced tools such as Gantt charts, allocation histograms and a reporting system - had the effect of 'blurring' the processed information.

Even with simple processes - where each planned task requires its execution, control, possible corrections and sending the results over to the designated workers - instead of one task in the classical project, the system generates a few. With more technologically complicated tasks or tasks requiring a number of steps, one task of the classical project would generate several or even a few dozen tasks.

The result is obvious. The initially small and simple project with every automatically added task generated from the previously defined process patterns grows larger and larger. Assuming that a single person performs only 5 tasks daily, the project, with a 10-person staff and after 50 work days, will contain 2500 tasks! Such a huge project, for its sheer size, becomes incomprehensible and escapes any analyses.

The missing relation between tasks

What is even more, there appears another and much more serious problem that can lead to many misunderstandings. When we are saying that every classical task is in fact a process, composed of a number of tasks and evolving as decisions are being made, we assume that there exists a clear relation between the task and process or processes involved in the task performance.

In the classical set of relations between tasks there are some very serious problems with adequate modeling the relations between the project task and the processes involved in it.

As the project execution starts, the system contains an approved project at certain level of generality and the process patterns 'awaiting' their initiation. Once the processes are initiated, the system, based on their defined patterns, automatically generates a network of micro-scale individual tasks and monitors their execution. Each process 'works for' the general tasks that have been previously defined and approved. The problem is that in the classical set of relations there is no relation representing this type of dependency.

The only available relation is that of 'superior task - subordinated task'. Without expanding the number of relation types we can proceed in two ways:

1) handle each project task as a summary task and place the tasks generated from procedures as their sub-tasks;

2) run processes 'alongside' the project tasks.

Both the approaches have serious drawbacks.

In the first approach, keeping the relation of 'processes for tasks' by setting up the project task as a summary task and generating process-related tasks as its sub-tasks, we maintain the designed project hierarchy, but at the very moment of running the process we affect its time- and cost- structure.

The summary task by definition is supposed to start exactly when the first of its sub-tasks is started and end when the last of its sub-tasks ends. The obvious lack of knowledge of all the decisions to be taken within the initiated process and the inability to predict the exact time of executing each of its individual tasks at the time of running the process precludes setting its exact end-time. As a result, the project task is shortened and the information on the planned course is irrevocably lost. Although we can, before initiating the process, make a snapshot of the project baseline and compare its future course to the snapshot, this proves to be impractical.

Aiming to keep the originally planned tasks, we can run the processes 'alongside'. In this situation, though the originally planned tasks remain intact, the connection between processes and tasks is virtual and is not represented in the system.

In the first approach we kept the structure but the project course monitoring has been prevented by overwriting the approved time- and cost- data with the process initiation parameters. The second approach can be described as feeding to the system completely independent and unrelated information, which displays more features of running independently two systems whose outcomes can be only compared and only if a strict discipline of data input is maintained.

The final conclusion seems to be as follows: although a process can be represented in the system in the very same way as a project, while using, however, only the classical set of relations between tasks, we cannot represent the relation between the project and processes involved in it.

Complex and partial tasks

Appropriate modeling of a situation where one task is executed 'for the sake' of, or within, another task can be achieved by introducing a relation which will enable connecting the project task with process tasks involved in it.

This relation is a connection between tasks, one of which will be called 'complex', the other 'partial'. One complex task can be connected with a number of partial tasks. Partial tasks do not add up to the complex tasks, instead they are handled as a sub-collection of actions involved in the complex task. The complex task, on the other hand, is not a direct sum of its partial tasks as it is the case with summary tasks, for partial tasks are only sub-specifications of individual elements of the complex task.

Example: I am planning to write an article in 10 hours total time. Then I go on to specify one of the task elements: I add a partial task 'Article drawings' and define that it is going to take 1 hour. The complex-partial relation ensures that the time of writing the article remains the same: 10 hours. The task 'Article drawings' will be called a partial task as it sub-specifies an element of the work involved in the complex task 'Write an article'.

The time- and cost- relations between the complex task and the partial task should be defined so as to adequately describe the actions taken in the partial task 'for the sake' of the complex task. Here are some basic ones:

The start date of a complex task is its planned real start date unless one of its partial tasks starts earlier. In that case, the start date of the earliest starting partial task is applied, with indication however to the difference.

1) The end date of a complex task is its planned end date unless one of its tasks ends later. In that case, the end date of the latest ending partial task is applied - at the same time alarming about the excess of the original assumptions.

2) The time effort needed for a partial task is the effort designed while planning the task, or the sum of efforts of all the partial tasks if it exceeds the effort planned for the complex task.

This model of relations enables us to define the expected time and work allocation in the micro-scale and compare the approved plan to its execution in the micro-scale.

Our projects, that a short while ago were growing beyond control, again assume the form of the classical project with the proper number and structure of tasks, while the processes take on actions 'for the sake' of the tasks, enabling us to compare at any time the values planned at the high level of generality to the data incoming in the course of the processes run 'for the sake' of the tasks.

Thanks to the complex-partial relation, we can plan and monitor processes executed in micro-scale with the help and fully taking advantage of the project management methodology, and at the same time observe the compatibility between the processes and the previously approved project.

Projects and processes vs. knowledge about 'what is going on' in the organization: metaprojects

Thanks to the connecting of projects to processes, we can efficiently and precisely monitor all the stages of their execution.

The project however is not the only structure organizing tasks. A large number of tasks executed in any organization are tasks unrelated to any other tasks. These can be for example all the entries in our organizers: what we should do and what we shouldn't forget about. The everyday duties, such as customer service or manufacturing, are not projects - instead they are handled by processes.

The question arises: what are the co-relations between day-to-day tasks and those resulting from projects? The very fact of collecting in one system of all the tasks and documents of the organization points at general questions referring to the operations of the whole company.

In many projects we can be executing tasks similar in their nature, based on the same technologies, related to similar problems or bringing similar financial effects for the company. Tasks generated by processes involved in day-to-day operations of the company can be also related to some projects, or their parts, affect the same factors, or refer to them in similar operational aspects.

Unfortunately, the classical structure of projects is oriented solely towards achieving goals without presenting the broader context of the operations carried out within the projects. Standard methods of collecting and storing data provide only information on: how often at the specified stage of the project the customer was called, how much time it took to overcome possible problems and how much time was devoted to administrating of the project course as such. Obtaining such knowledge is possible as each task can be assigned to a selected stage of the project, generated by a procedure or as a result of a decision, as well as have a category assigned from any number of category groups. These data however tell us nothing about the context of the initiated actions.

The problem can be defined in the following terms: what method of data classification should be adopted and on what basis should the registered data be co-related? What method can be accepted by the employees as 'natural' and comrehensible? Is there a method of classification that corresponds with the commonplace understanding of the work environment and the operations embedded in it and that can be recognized as standard enough to be used for the purposes of data analysis? How should the entering of data be organized so that preparing them for classification requires no additional work on the part of system users?

The problem seems to be complex, yet the solution to it proved to be unexpectedly simple. Let us note that if one asks any employee what exactly he is doing, he will be able to briefly list his duties and the nature of his individual operations; if one asks what he did last month, one will again get a short list of concise and well-defined subjects.

Answers to such questions are nothing else than the context of operations we have been looking for. Each individual, work goup, or organization as a whole executes its tasks with respect to their broadly defined oals or activity areas.

The solution then turns out to be the introduction of meta-structures similar to projects, called metaprojects. Metaprojects enable the employees to group their operations in line with self-defined classification criteria and thanks to it - registering the numerous contexts in which they are performed. In this way we introduce parallel grouping systems for activities and allow the users to place their tasks in each of these systems. What is more: we allow creating these structures by the individuals who will use them, which ensures better modeling in the system of the nature of each group's and its participants' operations.

For example: a manufacturing company can use these tools for creating metaprojects referring to products and technologies they involve. In a software company these can be metaprojects referring to product lines: 'Tilos', 'HumanWork', metaprojects referring to technologies: 'Data bases', 'Internet', 'Graphics', 'Animation' or metaprojects referring to day-to-day operations of the company: 'Customer service', 'Development ideas', 'Interface re-design'.

In meta-structures defined as above, a case that involves registering in the system a customer inquiry will be created in the metaproject: 'Customer service', meta-task 'Inquiries / Data transfer / ERP'; by the management it will be registered in the metaproject 'Contacts', in the structure 'HumanWork 2.0 functionality add-ons / Non-standard'; the software department will register it in the metaproject 'HumanWork', in the meta-task 'Team work / Managing resources / Consumable resource supply centers'.

This multitude of perspectives enables all the project participants to model in the system their own operational contexts, which significantly enhances access to the desired information.

Creating structures which enable placing the same processes in different contexts provides access to the most desired knowledge being a result of a multi-faceted analysis of the course of all the processes - i.e. knowledge on 'what is going on' in the organization.

Owing to this, we can learn e.g. how much time and what costs were allocated to the processes and projects in the context of the individual operations registered in the meta-tasks referring to the customer service. We can check what percentage of operations in the context of the customer service lead to signing contracts and to what products these operations were related. We can obtain information about what actions of the sales department result in signing new contracts and what actions generate large costs in the production departments or, finally, which contacts bring biggest profits.

This knowledge is impossible to get from standard time-line analyses of the process course offered by many workflow systems. The tools offered by such systems enable only technical improvements in the actions of a group of people. They can be used e.g. for cutting costs or clearing 'bottlenecks' that appear in the course of executed processes. Therefore, in part, they are rightly associated with the technicized automation of work.

HumanWork – the source of knowledge about the organization's operations

The problems discussed here are some of the many that required a detailed analysis and precise selection of solutions during the development of HumanWork.

The adopted solutions created an environment that enables all the collaborators to build up a real knowledge base about what they do and what results their actions bring.

This sort of knowledge has a fundamental significance from the point of view of every organization. As the source of this knowledge is not any theory but real data on real actions, it enables us to make right decisions on the directions for the development of the organization and the changes that need to be made in order to increase its efficiency.