published in: Proceedings of the International Conference on Engineering and Deployment of Cooperative Information Systems (EDCIS 2002), Springer Verlag 2002, LNCS 2480, pp 1-15

A Data Warehouse for Workflow Logs

Johann Eder, Georg E. Olivotto, and Wolfgang Gruber

Department of Informatics-Systems
University of Klagenfurt
A-9020 Klagenfurt, Austria
{eder,gruber}@isys.uni-klu.ac.at, georg.olivotto@aon.at

Abstract. Workflow Logs provide a very valuable source of information about the actual execution of business processes in organizations. We propose to use data warehouse technology to exploit this information resources for organizational developments, monitoring and process improvements. We introduce a general data warehouse design for workflow warehouses and discuss the results from an industrial case study showing the validity of this approach.

1 Introduction

Workflow management systems (WFMSs) improve business processes by automating tasks, getting the right information to the right place for a specific job function, and integrating information in the enterprise [8, 15, 1, 21, 2]. Workflow management systems support the execution of business processes as they require the definition of processes, automate the enactment of process steps and their execution guided by business rules and execution logic, and finally they document the execution of all steps of a business process. In particular, workflow logs [14, 17] contain the processing information for all instances of activities of workflow instances. Typically, they give account when which actor performed which task. So workflow logs contain very valuable information of the actual execution of business processes (as opposed of merely specified or desired descriptions of business processes). Thus they could be a very valuable resource for business process improvement, for reorganizations, and business process re-engineering. Workflow logs can also provide information for process controlling, and process monitoring. Workflow logs are also sources for information for process specifications and scheduling information like due dates, durations, or branching probabilities for conditional constructs (or-split).

Hunting for the treasures in workflow logs requires appropriate tools. Here we propose to use data warehouse technology and OLAP (online analytical processing) tools for aggregating, analyzing and presenting information derived from workflow logs. For further and deeper analysis, a data warehouse can also be used as base for data mining and knowledge discovery techniques.

Data Warehouses are structured collections of data supporting controlling, decision making and revision [22, 10, 12]. Data Warehouses build the basis for analyzing data by means of OLAP tools which provide sophisticated features for aggregating, analyzing, and comparing data, and for discovering irregularities. Data Warehouses differ from traditional databases in the following aspects: They are designed tuned for answering complex queries rather than for high throughput of a mix of updating transactions, and they typically have a longer memory, i.e. they do not only contain the actual values (snapshot data).

The most popular architecture for data warehouses are multidimensional data cubes, where transaction data (called cells, fact data or measures) are described in terms of master data hierarchically organized in dimensions. Transaction data, here the data about executing a certain instance of a certain workflow activity, is aggregated (consolidated) along these dimensions.

The OLAP-operations (drill down, roll up, slicing, dicing, etc.) allow analysts to rapidly derive reports in different levels of abstractions and view the data from different perspectives. Predefined reports deliver sophisticated management ratios and ad-hoc OLAP-queries allow to analyze the causes of deviations in these figures.

There is little work published so far on data warehouses for workflow logs. We are only aware of [4] where a data warehouse for workflow data for the HP process manager is described. The design goals were the simple installation and application of the Workflow–Data Warehouse which helps to detect critical process situations or quality degradations under different circumstances, e.g. different log sizes or different data loading and aggregation requirements. Our approach is more general since it is based on a general workflow meta model and explicitly collected queries which should be answered. Additionally, we were able to prove the concept with a prototype for a reasonably large workflow-log of an installation of the workflow management system @enterprise.

The rest of the paper is organized as follows: In section 2 we present our workflow meta-model which serves as basis for understand the structure of the workflow log and as as basis for the information supply. Section 3 presents the information demand side for the design of the workflow data warehouse in form of a set of interesting queries. In section 4 we present the architecture of our workflow data warehouse. In section 5 a case study is presented and discussed where our architecture was used on real data from workflow logs of a large workflow installment. Finally, in section 6 we draw some conclusions.

2 Workflow meta model

A workflow is a collection of *activities, agents,* and *dependencies* between activities. Activities correspond to individual steps in a business process, agents (software systems or humans) are responsible for the enactment of activities, and dependencies determine the execution sequence of activities and the data flow between them.

The meta model shown in Fig. 1 [6, 18] gives a description of the static and the dynamic schema aspects as well as of the organizational aspects. The workflow model we base our development of a warehouse architecture is able to capture workflows in different representation techniques, block-structured as well as un-

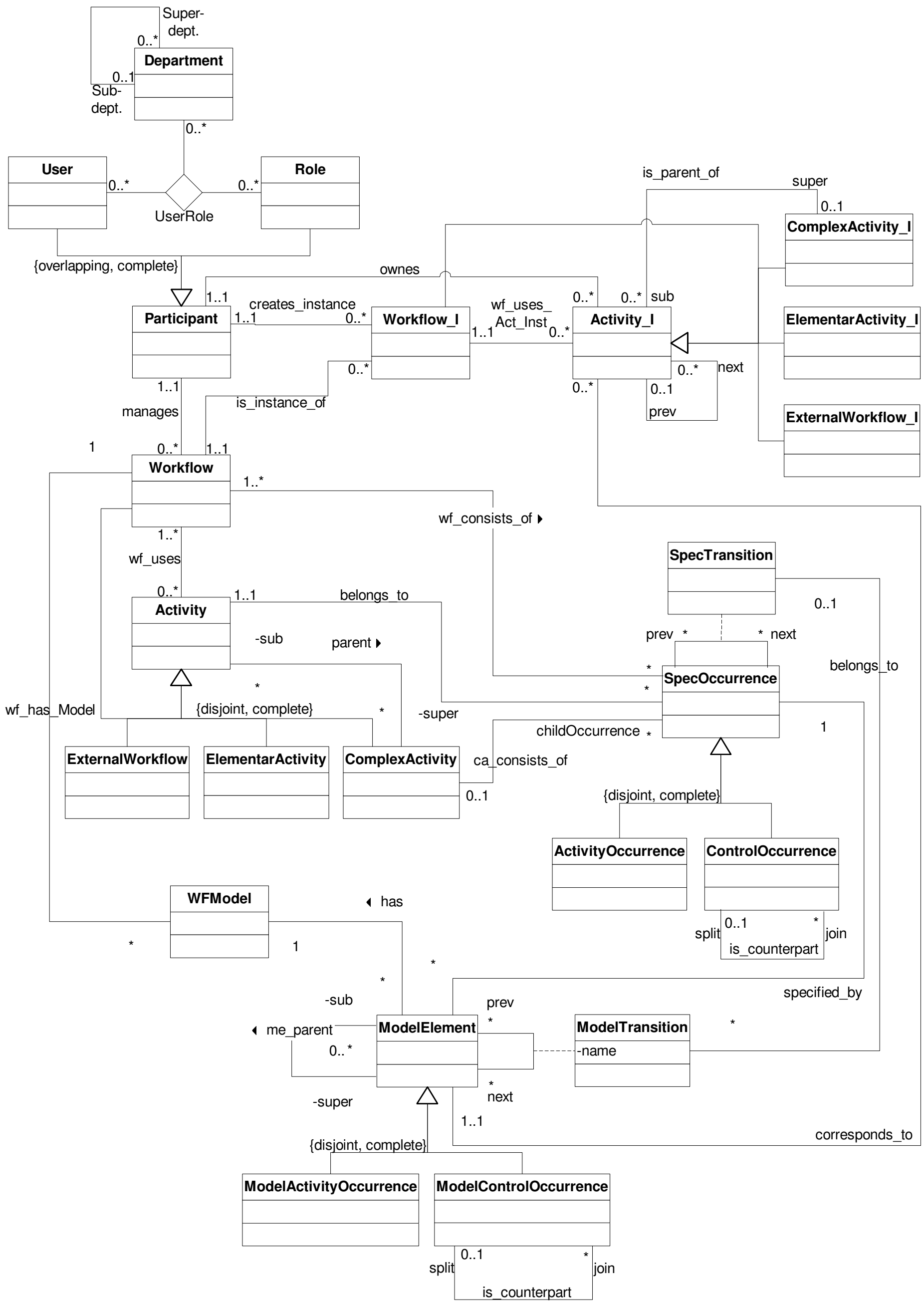

Fig. 1. Workflow meta model

structured workflows, text based programming language style representations as well as graph based representations.

The meta model consist of the following parts: The specification level contains the description of workflow types and activity types together with their composition structure via occurrences (see below). The model level contains the expanded workflow specifications, such that all activity appearances may have their individual characterizations (like due dates, agent, etc.). The instance level contains information about the execution characteristics of activities and workflow instances, and finally the organization level represents the agents, and the organizational structure of the company. We did not take the dimension of data into account, since there the differences between workflow systems are too big to allow an more general treatment.

In the following we want to describe briefly the notion of workflows, activities, occurrences and model elements used in the meta model [6]. A workflow consists of activities which are either workflows, external workflows, elementary activities or complex activities. Complex activities consist of other activities, represented as occurrences in the composition of the complex activity. The hierarchical relationship between activities, e.g. in which *parent*-activities an activity appears, is also declared. Within a complex activity a particular activity may appear several times, where every of those appearances can be unambiguously identified by the concept of occurrences. An occurrence is associated with exactly one activity and represents the place where an activity is used in the specification of a complex activity. Each occurrence, therefore, has different predecessors and successors, which is expressed by the association class *SpecTransition*.

There are two reasons for the distinction of an activity and its' (multiple) occurrence(s). The first is the possibility of activity reuse which means that an activity is defined once and can be used in several workflow definitions. The second reason is the simplification of maintenance which means that a subprocess has to be changed only once, and it is changed for all workflows where it appears. This allows that new workflows can easily be composed using predefined activities. Such a composition is also called a workflow specification. For the purposes of a workflow-log-data warehouse the identification of multiple appearances of the same activity is very important, as it allows to aggregate execution data of the same activity in different positions of the workflow and even in different workflows.

When a complex activity is used several times within a workflow, we also have to distinguish between the different appearances of occurrences, and the resulting elements are called model elements. Similar to the specification level, a workflow model has to be aware of its model elements and the hierarchical- and transition-relationship of those model elements.

On the instance level, the classes *Workflow_I* and *Activity_I* are used for the representation of the instances of workflows and their activities during runtime. Analogous to the workflow model, a workflow instance consist of activity instances, and for those, the predecessor- and successor-activity instances (if existing) are specified. Furthermore, a workflow instance belongs always to exactly one workflow (specification).

The *Participant* (agent or processing entity) is responsible for invoking workflows and the execution of activity instances. In our meta model, the participant can more precisely be modelled with the help of users and roles. The assignment of users to activities doesn't always take place directly. Ideally it is done by the usage of a role concept which allows a more generic user assignment [16] but in the meta model, presented in this paper, additionally a direct user assignment is also possible. Generally, we can express that users, roles or users in roles can participate in different workflows for different departments.

3 Requirements for Warehouse Architecture

The design of a Data Warehouse is determined by the available information in the workflow management system and by the information needs of the decision makers and the analysts. Therefore, each workflow warehouse will need adaption to the peculiarities of a certain enterprise information system. Here we introduce a prototypical data warehouse architecture for workflow histories based on prevailing standards of workflow systems on one hand, and on typical information needs of process managers and analysts. In the next section we will then show how this general data warehouse architecture can be adjusted to a particular application of a particular workflow management system.

For the information supply we build on the workflow metamodel we presented in the previous section. For the information requirements we should be able to answer the following questions

- who are the users of the system
- which data are required
- which queries should be answered

The users could be both the administrators (= system administrators, workflow modellers etc.) and the users of workflow systems. The relevant components of workflow systems are the workflows, their activities, the participants and the used servers. The formulation of a set of queries is of particular importance for the design of the Data Warehouse structure and is given in the following [18]:

- **Query 1**: how often are particular workflows enacted (the information provided by this query could be used to identify core processes which should be preferred in optimisation and priority assignment);
- **Query 2**: which activities within these workflows are enacted (with this query preferred navigation paths through workflows can be identified);
- **Query 3**: the enactment of which workflows regularly lead to deadline misses;
- **Query 4**: the enactment of which activities lead to these deadline misses;
- **Query 5**: what is the amount of these deadline misses;

- **Query 6**: how often are the workflows, which cause deadline misses, enacted (with this information one can decide if it is worth to optimise the workflow or not);
- **Query 7**: how many different users participate in the enactment of particular workflows;
- **Query 8**: who are the users participating in activities which lead to deadline misses frequently;
- **Query 9**: who are the regularly overloaded users and who have regularly free capacities;
- **Query 10**: which users participate in which workflows;
- **Query 11**: comparing measures of different versions of a process;
- **Query 12**: comparing a process–type during several periods of time;
- **Query 13**: analysing the executions of an activity in different processes;
- **Query 14**: analysing the executions of an activity by different users;
- **Query 15**: monitoring the performance of users during several periods of time;

Another important component is time [18]. Various analysis of time aspects could provide valuable results. The detection of peak times could probably help to solve the problems detected by queries three and four by enhancement of the time management. This could help to avoid expensive process optimisations. The detection of peaks could also permit a basis for the dynamic distribution of the load on different servers.

4 Warehouse Model

Based on the requirements in the previous section we introduce now the design of a hypercube (fig. 2) that allows us to answer all the queries formulated above. It consists of the following six dimensions:

- workflows
- participants
- organisation
- servers
- time
- measures

The dimensions of the workflows and the participants result directly from the meta model presented in figure 1. The occurrences, which are necessary for reuse purposes, are the level of granularity (LoG) in the workflow dimension. Starting from here there are different alternatives to build up the structure of the workflow dimension. One possibility is the separation of the workflows and the activities in two different dimensions to cope the problems arising from the reuse of particular activities in different workflows. The disadvantage of this approach is that there is no chance to drilling down to the occurrences starting from the workflows. Another alternative is to consolidate the occurrences to the relevant

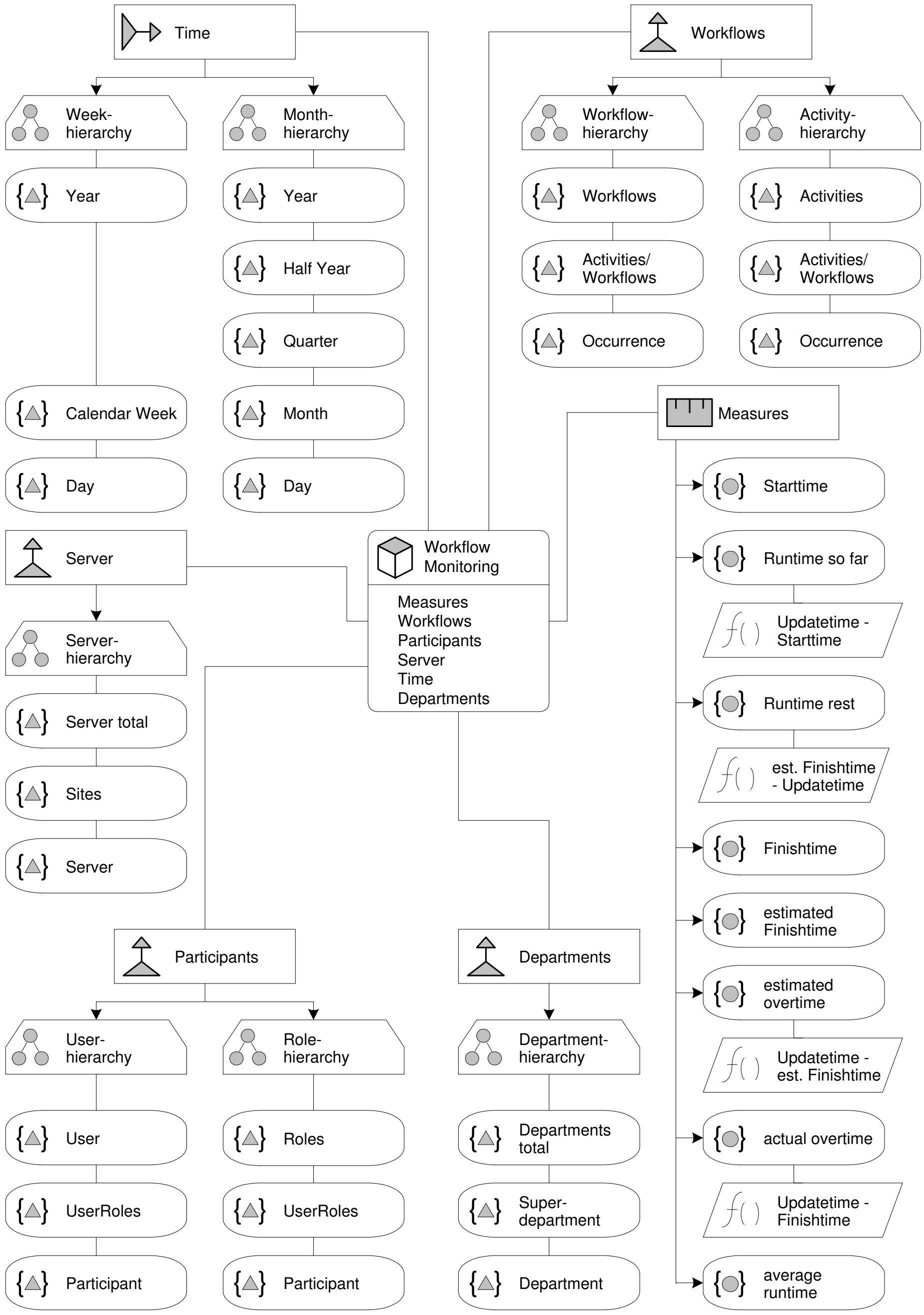

Fig. 2. Data Warehouse schema

activities and to consolidate the activities to the corresponding workflows. The problem that occurs in this solution is that every activity can belong to different workflows and their values would have to be prorated to different workflows.

Our solution is to consolidate the occurrences in the first step to elements which represent a combination of the corresponding activities and workflows. These elements can be consolidated either to the workflows or to the activities. This approach allows the drill down to the occurrences starting from the workflows and prevents an ambiguous assignment of activities to workflows. In the Data Warehouse the benefit of the consideration of reuse aspects is given by the possibility to compare the occurrences of an activity appearing in different workflows.

For the representation of the participants and the departments there are different alternatives too. In the first we consider the separation of the users and the roles into two different dimensions whereby the departments are built by the consolidation of the users. The meaning of this solution is that every user can act for his department in any role. In practice this solution is insufficient because it is not unusual that a user works for different departments. In our approach we separate the participants and the departments in two different dimensions. The lowest level of the participant dimension is represented by the participants themselves which are consolidated to elements which are a combination of users and roles. In the next step these elements are consolidated either to the users or to the roles.

The reason for the separation of the users and the departments in two different dimensions is the fact that users can participate in different workflows for different departments. Thus it is not possible that a department is represented as a consolidation of its' members.

The introduction of a server dimension has technical reasons and provides the possibility to get information about the load of the different servers which could be used as a basis for a distribution of the load. The structure of the server dimension is very simple and self–explanatory.

In the time dimension we selected a calendar day which consolidates to months, quarters and years as a chronon [13]. The consolidation to calendar weeks must be done in a separate hierarchy because of the fact that in different years a week could belong to different months.

For the measure dimension we selected the following set of measures:

- starttime (used to determine points of time on which activities usually are started)
- runtime elapsed
- runtime rest till deadline
- finishtime
- estimated finishtime
- estimated overtime
- actual overtime
- average runtime

DEPT	*Total Departments*	
ITEMKIND	*Total Itemkinds*	
ACTIVITY	*Total Processes*	
AGENTS	*8590147888*	
	UNIT	MEASURE1 ...
	Number	Days
TIME ...	started	Duration
12/1999	118,00	76,28
1/2000	376,00	768,98
2/2000	472,00	1.056,91
3/2000	455,00	1.074,60
4/2000	270,00	835,37
5/2000	265,00	767,23
6/2000	310,00	606,53
7/2000	731,00	669,94
8/2000	471,00	508,48
9/2000	842,00	429,37

Fig. 3. example of analysis 1

The throughput, the turnaround time and the resource consumption could be further examples for measures. The Data Warehouse schema contains information about the build time properties as well as the run time properties of the workflows. The build time properties are given in the dimension structures and the run time properties are represented by the data values.

Additionally we want to emphasize the different meanings of the term time as a dimension and as a measure. The relevance of a time dimension is fundamental for a Data Warehouse and is already part of its definition [11]. It is used to represent data which belong to others than the actual point of time. As a measure we need time e.g. to get information about the execution duration of activities, to determine points of time at which usually many instances are created or to detect deadline violations.

5 Case Study

The goal of the case study was the prototypical implementation of a Log–Data Warehouse to analyze the validity of the theoretically elaborated concepts. The source workflow system was @enterprise of Groiss Informatics GmbH, Klagenfurt, (www.groiss.com) with a large number of installations in German speaking countries. @enterprise is partly based on the prototype workflow system Panta Rhei [7] which was developed at the University of Klagenfurt.

For this case study we gratefully received real data from a large organization on basis of anonymity. The data stems from the workflow log of the time period from December 1999 to September 2000. The workflow implementation

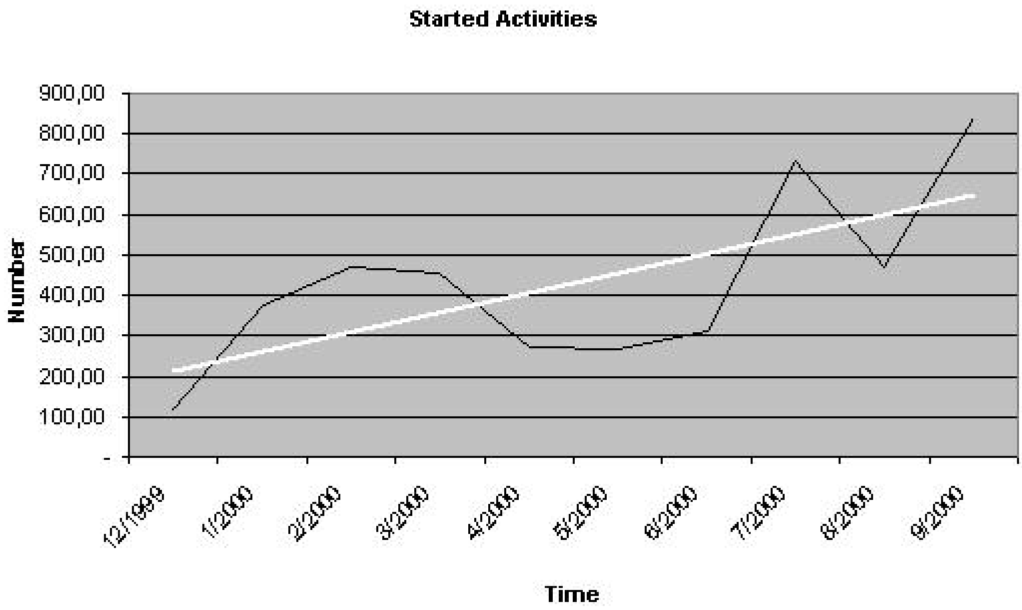

Fig. 4. example 1: started activities graphical

contained, at the time of the case study, 22 different workflows, 325 different activities, 2.740 occurrences, ∼171.100 instances (∼5.300 workflow instances and ∼165.800 activity instances), ∼1.000 users and 75 roles.

The following adaptions to our general workflow model were necessary: The workflow descriptions contain neither duration information nor due dates. Therefore, all measures requiring this information and all queries based on these data had to be abandoned. With the resulting multidimensional system, several of the queries formulated in the previous section can be answered easily. The adaptations of the Data Warehouse structure in the case study were made to adjust to the specifics of the particular workflow model of @enterprise and its specific log structure.

Experimenting with the workflow data warehouse we soon found out that the results of the given queries quickly rose other queries and lead to new information demands. Several of those queries could again be easily answered with our prototype system, e.g.:

- activity–transitions concerning the number of transitions and the time consumed by the transitions;
- determining the types of activity termination (cancelled, finished or compensated) in absolute and relative numbers;
- activity runtimes;
- determining how long activities are in the worklists of different users;
- monitoring the evolution of the number of instances of different activities;
- determining how many instances are created by different users;

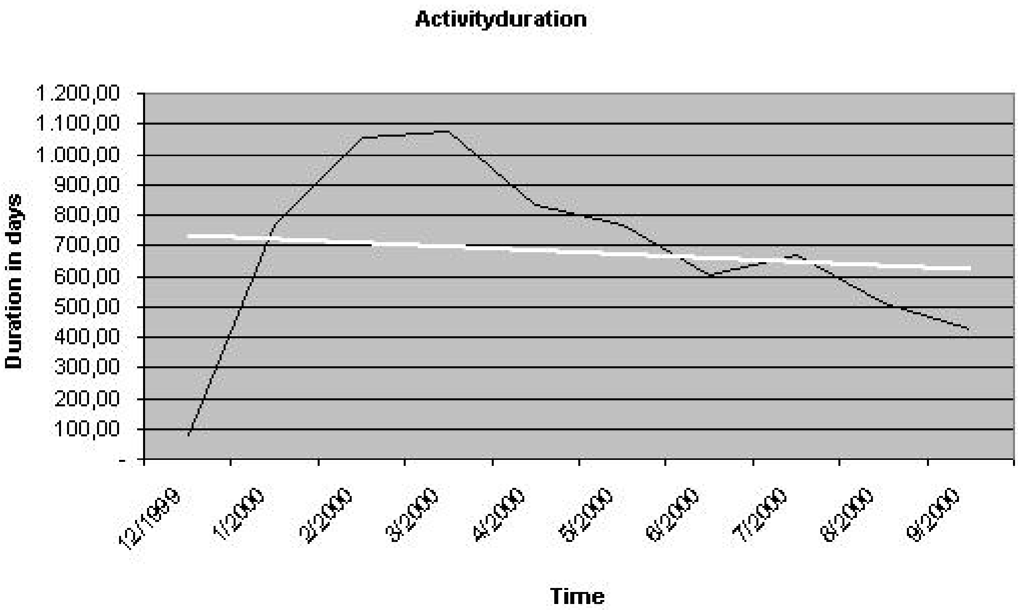

Fig. 5. example 1: Activity durations graphical

In the first example we want to analyse the correlation of the number of the instances created by agent 8590147888 and the times the activities are in his work list. Figure 3 contains the source data. The first line of the table denotes that in December 1999 a total number of 118 instances were created by agent 8590147888 and the sum of the times the instances were in his work list was 76.28 days. The remaining lines can be interpreted analogically. The black line in figure 4 represents the number of created instances for the particular months the white line shows that the number of created instances on the average is growing. Figure 5 shows how long the instances were in the work list of agent 8590147888. Again the black line shows the values for the particular months. The white line represents the falling trend. The conclusion of this example is that although the number of created instances is increasing the processing times of the instances are decreasing. Two possible interpretations could be that either the agents' experience is increasing continuously or the increasing number of instances raises the agents' efficiency.

For the second example shown in figures 6 and 7 we use two computed measures: The *average runtime* of occurrences is computed by dividing the sum of the occurrence runtimes through the number of their instances. The *performance measure* relates the average runtime of an occurrence executed in a particular department with its overall average. Figure 6 shows that department 8590147029 has a bad performance in executing occurrence 8590170011. Interested in the performance of the other occurrences executed in this department we could easily transform the table in figure 6 to the one presented in figure 7.

AGENTS	*Total Agents*	
ITEMKIND	*Total Itemkinds*	
TIME	*Total Years*	
ACTIVITY	*[illegible]*	
UNIT	*Number*	
	MEASURE1	
DEPT	avg. Runtime	erformance measure
Total Departments	333.521,39	100,00
8589934611	321.541,25	96,41
8590147009	260.948,00	78,24
8590147029	437.008,15	131,03

Fig. 6. example of analysis 2

6 Applications

The major application of the data warehouse for workflow logs is to support the business process improvement cycle. The analysis of the workflow execution data should provide insight for possible improvements in duration, throughput, adherence to deadlines, resource consumption, etc. The Data warehouse also provides statistical data for business process simulation. If business process reengineering leads to new workflow definitions, the execution of these new workflows can be compared with the old versions. Furthermore, benchmarking of workflow execution is supported on a very detailed level.

The data warehouse can also be used as a basis for data mining, for e.g. for fraud detection, for finding typical execution patterns, etc. It can also be used to derive key quality figures for measuring the performance and quality achievements of organizational units.

If the log data is fed into the log-warehouse shortly after their creation, the warehouse also supports process monitoring in an excellent way. Here an operational data store as intermediary between the workflow management system and the log warehouse could be very helpful. Such an architecture has been successfully applied for controlling and monitoring industrial production processes, e.g. in the semiconductor industry. Comparisons between planned execution and actual performance can constantly be monitored by predefined OLAP reports and workflow controllers can benefit from an improved overview.

Another important area of application is the application of the data warehouse approach for workflow logs for process mining. For this purpose, the log is used for discovering the business rules and the workflow models which lead to the execution patterns which are observed.

Data mining in the context of workflow logs to discover information about the workflow instances of various kinds is addressed e.g. in [3, 5, 9, 19, 20]. Different methods have emerged, targeting different kinds of data, such as workflow control structure, resource allocation, time consumption parameters and so on. The

AGENTS	*Total Agents*
ITEMKIND	*Total Itemkinds*
TIME	*Total Years*
UNIT	*Number*
DEPT	*8590147029*

	MEASURE1			
ACTIVITY	started	avg. Runtime	erformance measure	started %
17179955566	4,00	3.219.215,00	616,40	12,12
17179966180	46,00	104.547,39	337,19	9,96
17179966215	40,00	225.814,23	200,14	8,77
8590169987	40,00	375.113,68	171,62	12,90
17179972609	27,00	352.891,85	170,90	6,31
8590169977	48,00	233.363,44	169,23	14,16
8590169977	48,00	233.363,44	169,23	14,16
8590170011	27,00	437.008,15	131,03	11,25

⋮

Fig. 7. example of analysis 2 - detail

specialized methods take advantage of the nature of the specific data to which they are applied. With the results, reverse engineering of the process specification can be performed to improve the system as a whole.

Compared to these approaches we emphasized application areas where the workflow process model is known. Nevertheless, the proposed warehouse model can also store log data of ad-hoc workflows and may thus serve as a basis for process mining techniques mentioned above. The focus of our work is to exploit the workflow log and building a data warehouse to obtain aggregated information such as i.e. to detect critical process situations or quality degradations under different circumstances, rather than re-engineering workflow specifications form the log. However, these process mining techniques can deliver important data for discovering typical execution scenarios, dependencies between decisions and probabilities of workflow instance types. Business process re-engineering and workflow improvement will benefit form a combination of the approaches.

7 Conclusions

How an organization really works is precisely documented in the logs of its workflow management system. These workflow histories, therefore, are a very valuable source of information for a multitude of applications, from workflow monitoring and controlling to business process re-engineering, from statistics generation to fraud detection.

For exploiting these valuable data we propose data warehouse technology. Most workflow management systems provide already some functions for analyzing and browsing workflow-logs. We see the main advantage of a data warehouse based systems as follows:

- Sophisticated OLAP tools can be used for analyzing the data. These OLAP tools offer much higher functionality and are higher optimized than typically monitoring interfaces to workflow systems.
- Workflow execution data is typically only a part of an information system and decision support infrastructure. Using warehouse technology allows seamless integration of these data.
- Frequently, larger corporations employ several workflow management systems for the execution of different business processes. The log-warehouse approach offers integrated and/or comparative analysis of the data produced by the different systems.
- Using an independent log-data warehouse allows also to measure the success of a change (replacement) of workflow management systems.
- For analysis and benchmarks workflow execution data has to be augmented with other data from different information systems of the organisation or data from external sources. Again, using warehouse technology opens this option for decision support.
- Last but not least, the separation of operational process support and analytical decision support is mostly a performance improving architecture. Moreover, warehouses typically can hold data for longer periods of time and thus are more suited for trend analysis, etc.

We developed a very general architecture of a log-data warehouse based on a workflow metamodel and typical information needs of process managers. In a large case study with data from an actual workflow installation we could show that the approach is viable. It was interesting to witness how the proposed model could be used to rapidly generate unexpected reports, generate various comparisons and trend analysis. Here the power of OLAP tools could be used very efficiently.

References

[1] Work Group 1. Glossary: A workflow management coalition specification. *Workflow Management Coalition, Brussels, Belgium*, V 1.1 Final, 1994.

[2] Work Group 1. Interface 1: Process definition interchange. *Workflow Management Coalition*, V 1.1 Final(WfMC-TC-1016-P), October 1999.

[3] Rakesh Agrawal, Dimitrios Gunopulos, and Frank Leymann. Mining process models from workflow logs. In Hans-Jörg Schek, Fèlix Saltor, Isidro Ramos, and Gustovo Alonso, editors, *Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology, Valencia, Spain, March 23-27, 1998, Proceedings*, volume 1377 of *Lecture Notes in Computer Science*, pages 469–483. Springer, 1998.

[4] Angela Bonifati, Fabio Casati, Umeshwar Dayal, and Ming-Chien Shan. Warehousing workflow data: Challenges and opportunities. In *The VLDB Journal*, pages 649–652, 2001.

[5] Jonathan E. Cook and Alexander L. Wolf. Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology*, 7(3):215–249, 1998.

[6] J. Eder and W. Gruber. A Meta Model for Structured Workflows Supporting Workflow Transformations. In Yannis MANOLOPOULOS and Pavol NAVRAT, editors, *Sixth East-European Conference on Advances in Databases and Information Systems, ADBIS 2002, Bratislava, Slovakia, September 8-11, 2002, Proceedings*. Springer, 2002.

[7] Johann Eder, Herbert Groiss, and Walter Liebhart. The workflow management system panta rhei. In Asuman Dogac, Leonid Kalinichenko, M. Tamer Ozsu, and Amit Sheth, editors, *Advances in Workflow Management Systems and Interoperability, 1997*, pages 129–144.

[8] Dimitrios Georgakopoulos, Mark F. Hornick, and Amit P. Sheth. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, 1995.

[9] Joachim Herbst and Dimitris Karagiannis. Integrating machine learning and workflow management to support acquisition and adaption of workflow models. In *DEXA Workshop*, pages 745–752, 1998.

[10] B. Hüsemann, J. Lechtenbörger, and G. Vossen. Conceptual Data Warehouse Design. In *Proc. of the International Workshop on Design and Management of Data Warehouses (DMDW 2000)*, Stockholm, 2000.

[11] W. Inmon. *Building the Data Warehouse*. John Wiley and Sons, New York, 1 edition, 1992.

[12] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis. *Fundamentals of Data Warehouses*. Springer-Verlag, 2000.

[13] C. Jensen. A consensus glossary of temporal database concepts. In S. Jajodia O. Etzion and S. Sripada, editors, *Temporal Databases: Research and Practice*, pages 367–405, 1998.

[14] Pinar Koksal, Sena Nural Arpinar, and Asuman Dogac. Workflow history management. *SIGMOD Record*, 27(1):67–75, 1998.

[15] P. Lawrence. *Workflow Handbook*. John Wiley and Sons, New York, 1997.

[16] W. Liebhart. *Fehler- und Ausnahmebehandlung im Workflow Management*. Dissertation, Klagenfurt, 1998.

[17] Peter Muth, Jeanine Weisenfels, Michael Gillmann, and Gerhard Weikum. Workflow history management in virtual enterprises using a light-weight workflow management system. In *RIDE*, pages 148–155, 1999.

[18] G. Olivotto. *Ein Data Warehouse zur Analyse von Workflows*. Master Thesis, University of Klagenfurt – Department of Informatics–Systems, 2002.

[19] A.J.M.M. Weijters and W.M.P. van der Aalst. Process mining. discovering workflow models from event-based data. In *Krse, B., De Rijke, M., Schreiber, G, and Someren, M. van (Eds.), Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence*, pages 283–290, 2001.

[20] A.J.M.M. Weijters and W.M.P. van der Aalst. Rediscovering workflow models from event-based data. In *Hoste, V., and De Pauw, G. (Eds.), Proceedings of the Eleventh Belgian-Dutch Conference on Machine Learning*, pages 93–100, 2001.

[21] Workflow Management Coalition. The Workflow Reference Model, 1995.

[22] M. Wu and A. Buchmann. Research Issues in Data Warehousing. *BTW'97*, 1997.

