2

Usability of Some Workflow Products in an Inter-organizational Setting*

J. Juopperi, A. Lehtola, O. Pihlajamaa, A. Sladek, J. Veijalainen
VTT Information Technology, Multimedia Systems
P.O.Box 1203, FIN-02044 VTT, Finland.
Telephone: +358-0-4565953. Fax +358-0-4567028.
Email: {jari.juopperi,aarno.lehtola,olli.pihlajamaa,
jari.veijalainen,aija.sladek}@vtt.fi

Abstract

Workflow techniques have gained plenty of attention lately as a means to support business process re-engineering. One has also argued that they are an important asset when integrating legacy systems within an organization. In the following, their applicability in an inter-organizational setting will be evaluated. This is done by grasping requirements of three different inter-organizational applications and relating them with the facilities offered by some existing workflow products, notably IBM FlowMark, Staffware, and TeamFlow. The reference architecture and terminology used in comparisons is that of the Workflow Management Coalition. The focus in this context is on specification needs deducible from the applications and the corresponding support by the products. Further point of interest is the applicability of the products as an integrating facility of legacy systems within and between organizations. A third point of interest is the support for advanced transactional properties which is of prime importance in the light of the application analysis in these environments. The overall conclusion is that the products analysed lack many of the advanced transactional features and specification facilities needed in inter-organizational environments, though some products will have more of them in their future versions.

Keywords

Workflow, workflow tools, computer supported cooperative work, business process re-engineering

*This work was done in the ESPRIT LTR project TransCoop (EP8012) which is partially funded by the European Commission. The partners of TransCoop are GMD (Germany), University of Twente (The Netherlands) and VTT (Finland).

A. Sölvberg et al. (eds.), *Information Systems Development for Decentralized Organizations*

1 INTRODUCTION

Workflow techniques were first developed for and used in distributed project management systems. The background were big government funded software projects in USA in the seventies [Gru94]. Currently, they are seen as the key technique to facilitate business process re-engineering, because they support process-oriented work-groups. The tools needed are part of a new set of products and resources for solving business problems [Loc94].

We are here interested especially in inter- and intra-organizational workflows and systems which could support them. Such workflows span between autonomous organizations and fairly autonomous* organizational units. These workflows might be fully automated, in which case no human beings are involved in the execution of the steps (or activities, see below the terminology), or they might contain manual steps. In both cases, correctness of the workflow process definitions, recovery from all kinds of errors, as well as intra-instance and inter-instance concurrency are important issues which should evidently be addressed by the supporting systems. Below we will show, based on three real-world cases, that these *transactional system requirements* are really adequate. We will also discuss other system requirements deducible from the cases and evaluate three products against all the requirements so gained. The cases are the PortNet system used by Finnish ports and the National Board of Navigation (NBN)†, the 101-service provided by Telecom Finland, and a software error reporting system used by the Valmet Corporation. The analysed products are Flowmark by IBM, Staffware by Staffware PLC and TeamFlow by ICL Personal Systems.

The workflow field is immature, which is also reflected in the lack of widely accepted common terminology. In this context we will apply the architectural approach (see figure 1) and terminology defined by the Workflow Management Coalition (WfMC) [Wor94]. The WfMC reference model distinguishes between manual and computerized parts in (business) processes. From the modeling point of view, the phases identified are the *process definition* and the subsequent *instantiation* of the process definition, again and again, to perform the actual work. It is foreseen that the process definitions will be modified from time to time. In this terminology, *a process* is a finite collection of manual or computerized *activities* organized to accomplish some business process. *Workflow activities* (processes etc.) are computerized steps, whereas *manual activities* (processes etc.) are non-computerized steps. Each performs a piece of work.

From the point of view of the global process definition, semantics of an activity are indivisible and well-defined. At the other level of the process definition, e.g. within the local environment, however, there maybe a complete process definition again, which corresponds to the activity. In this case it is called *sub process.*

There is no much related work known to us where application analysis, the WfMC reference model and products would have been brought into relation with each other in the way we have done it here. There is, however, a lot of work done in other contexts towards advanced workflow concepts.

Klein [Kle91] has shown that most known *transaction dependencies*, which seem to be

*In this context we do not handle further the definition of autonomy and its consequences. The interested reader is urged to consult e.g. [Vei90, VEH92, Vei92, Vei93]

†We use below also the Finnish acronym MKH

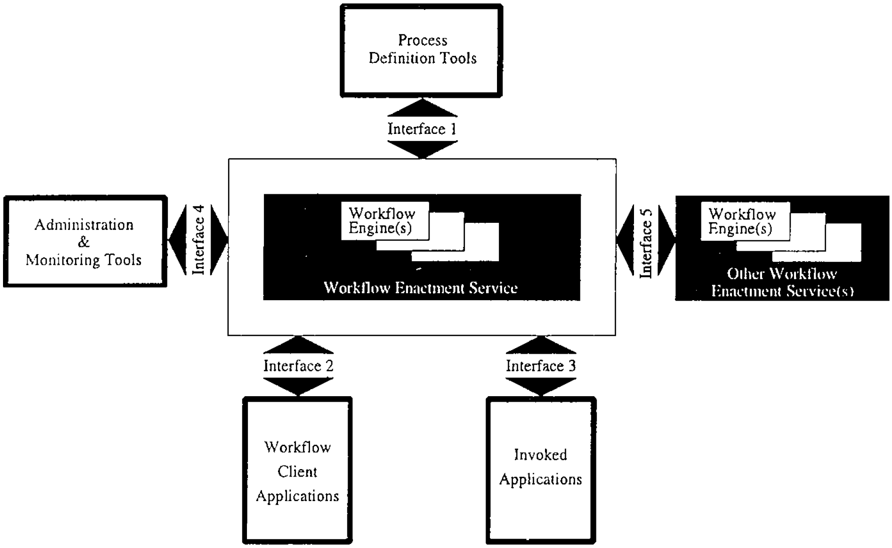

Figure 1 The WfMC reference model

important in the light of our application analysis, can be represented already by Create-, Finish-, Commit-, Abort-, Compensation- and Weak Commit dependency.

In general, if the integrity constraints on the persistent data do span multiple systems, then concurrent workflows may interfere with each others in an undesirable manner and thereby there is a need to control concurrent workflows. One of the most relevant contribution towards this end is the concept of*transactional workflows*[GH94] which include a workflow specification language and which explicitly address the problems of concurrent workflows. The *Contract Model* [Wäc91, WR92] introduces a control mechanism over so-called ACID transactions [GR93] by invariant predicates which can be specified to hold across tasks, and which prevent possible interference of concurrent workflows. In the approach presented in [BDS+93] a workflow specification model [ASSR93, SR93] and an open nested transaction model [WS92] is combined. In their approach the workflow enactment service above would enforce termination dependencies and the control and data flow dependencies within a workflow definition. The semantic transaction management component they propose would manage the compensation dependencies, and the dependencies between incompatible tasks of different workflows to ensure multi-system consistency.

Transaction metamodels provide a common framework within which one can specify and reason about the nature of interactions between transactions in a particular model. ACTA [CR90b, CR90a, PC94] allows one to specify transaction dependencies as well as the so-called conflict relations of operations.

The rest of this paper is organized as follows. In Section 2, the requirements from the applications and literature are discussed. In Section 3, we relate the requirements with the three products and analyse their deficiencies. Section 4 concludes the paper.

2 CASE STUDIES

2.1 The PortNet case

PortNet [Tra94] is an EDI system to handle notices of vessel arrivals and departures. Its goal is to simplify the current business procedures of the user organizations, to create a uniform business process model for all Finnish ports and to gain reduced cost as well as other benefits.

The overall system structure and main functions

The environment of the PortNet system is shown in figure 2. The information exchange between various parties (the interface 5 in figure 1) is standardized partly through a few EDIFACT message types. Those parts of the process definition which go beyond the data formats are given in natural language.

When a foreign ship visits a port, various activities must be carried out by a number of commercial and administrative organizations. For example, the ship arrival must be declared to the customs, the information concerning the arrival date and the ship itself must be recorded into various databases, navigation of the ship and other services must be ordered etc. The activities related to the ship departure include the recording of information concerning the departure date, navigation and other services, calculating the duration of the stay of the ship and the related cost etc. Next we outline the processing of the arrival of a ship.

There are essentially seven types of actors involved in the arrival of a ship: a ship, an agent, a stevedoring company, customs, a port, the National Board of Navigation (NBN) and a sea pilot station. An agent is a company which organizes the handling and further delivery of the cargo of the ship. The National Board of Navigation is an authority which controls the traffic near the coast of Finland.

The process definition describing the arrival of a ship is instantiated when a local agent receives a manifest concerning the arrival of ship from a foreign agent. When this happens, the local agent informs the selected stevedoring company, customs and the local port about the forthcoming arrival of the ship. The local agent also stores the received service order into a service database. Finally, the local agent creates an advanced arrival notice (AAN) and sends it to the port and the NBN. The AAN contains detailed technical information about the ship, e.g, name, radio call sign, length, width, machine power etc.

When the local port receives the AAN from the agent or from the NBN, the contents of the AAN is stored into a database and the local service companies are requested to prepare to handle the ship. When the NBN receives the AAN from the agent, it transmits the AAN to the appropriate sea pilot station so that a pilot can be reserved to navigate the ship to the port.

The agent may receive further notification from the foreign agent from time to time because the information contained in each notification is only an estimation and may have to be refined repeatedly. For example, the exact date of arrival of a ship is affected by many unpredictable factors, such as the weather conditions on the sea, the possible delay in the intermediate ports, etc. As the exact arriving date draws closer, the information will become more accurate.

When the ship arrives at the port, the actual arrival time (ATA) is stored into the local

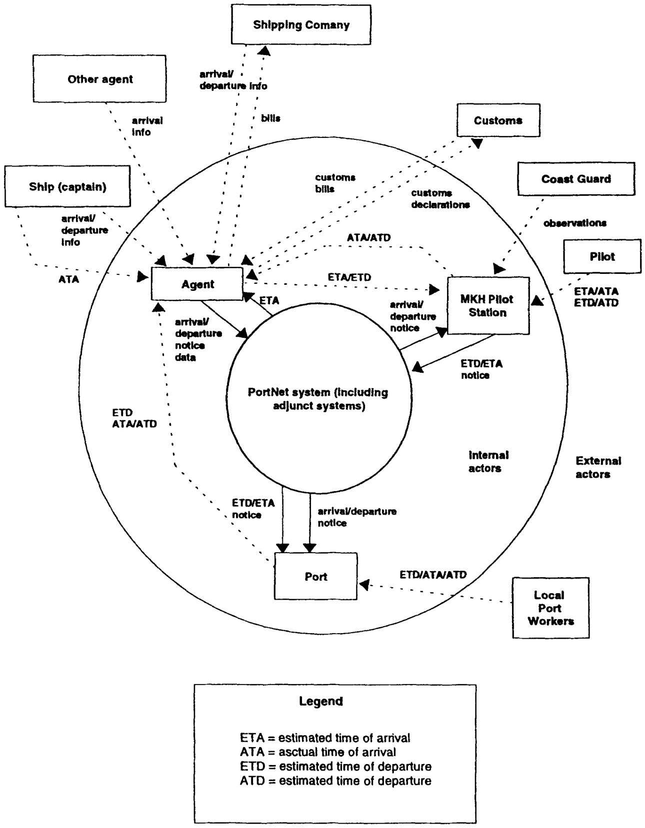

Figure 2 Handling a ship visit: environment diagram

database and the NBN is informed about it. The agent asks the ATA (currently by phone) and then creates a Final Arrival Notice (FAN), stores the real arrival date of the vessel into the database and sends the FAN to the port and to the NBN. It is required that this process finishes no later than three days after the arrival of the ship.

Observations from the application

Next we summarize the characteristic features of the PortNet case. First, the processes in the environment are complicated in structure and span several organizations. This requires the interface 5 in figure 1 to be carefully designed and the overall process definition must span several organizations. Second, the processes have a long duration, as the ships move slowly; at the same time, also deadlines are needed, so the enactment service must support workflows with long duration and deadlines. Third, the environment is heterogeneous in many respects (different kind of processes at different participants, very different equipment, manually mediated information flows, like captain confirming the order of a sea pilot by phone or an agent asking information) and includes many autonomous participants. These aspects must be taken into consideration when the process definition tools and workflow engines are designed to interoperate.

Fourth, the environment shows clearly the need of a hierarchy of the processes; The port bases its functioning on the information of the ship arrivals and departures, but the small piece of information telling that the ship is coming/leaving starts a big internal process within the harbour. The same applies to other participants. Finally, a ship may call off its visit to a particular harbour and go to another, e.g. due to ice barriers or storms. This causes the need to reverse some activities already done in harbour. In transactional sense one needs compensation.

The PortNet case would require the process definition tools to allow the creation of definitions which contain both manual and computerised activities. They should also support hierarchical definition of the workflow processes. This is needed to help a process modeller to use top-down approach and to promote reusability.

Due to the hierarchical nature of processes tools should support the description of two kinds of activities: *basic activities* and *process activities.* Basic activities are leaf-level activities that are not structured but the activities where the actual work is done. Process activities are executable workflow processes used as subprocesses.

The support for *transactional properties* is needed for correct and reliable execution of workflow process instances. Workflow tools should also support different subsets of traditional transactional properties for different activities and processes. The required set of supported properties varied from empty to full ACID-properties[GR93]. For example, the AID-properties were needed for some activities, such as "Checking, approving and sending the notice" performed by the Agent. On the other hand, the Agent's activity "Checking minimal data requirements" had no transactional properties and it could be interrupted at any time. At the upper levels, the AD-properties were required for the inter-organizational scenario, "Handle a ship visit" (see PortNet case 2.1).

Support for long-living process instances is also needed. For example, the execution of the whole PortNet inter-organizational process can take up to two months. During the execution time, other process instances may need the partial results. Thus a workflow management system should export the updates of a workflow process instance before its completion. *Backward* and *forward recovery* is needed in case of failures.

Activity assignment should be allowed to single users, to roles and even to organizations. In some cases assignment must be dynamic and thus it must be solved at run time.

A workflow management system should offer three basic operations for the control of workflow process activity instances: `Begin` which starts an execution of the activity instance, `Release` which finishes an execution of the activity instance and `Cancel` that cancels an execution of the activity instance because of a semantic failure or an user initiated cancel operation.

Describing the different options of control flow (sequential/parallel execution of the activities, splitting/joining the threads, time dependencies between activities, etc.) is a tricky problem. One has found it useful to use *dependencies* between activities to express them. Ordering dependencies form the basis of the control flow definition (routing) of the workflow process. In addition to *serial dependency*: `Release` $\mathcal{A} \rightarrow$ `Begin` $\mathcal{B}^{\ddagger}$ many other *ordering dependencies* between the activities $\mathcal{A}$ and $\mathcal{B}$ and their basic operations should

‡The execution of $\mathcal{B}$ will be started after the execution of $\mathcal{A}$ has been successfully finished, i.e. released.

be supported by the workflow tools: `Cancel` $\mathcal{A} \rightarrow$ `Begin` $\mathcal{B}$, `Release` $\mathcal{A} \rightarrow$ `Release` $\mathcal{B}$, `Release` $\mathcal{A} \rightarrow$ `Cancel` $\mathcal{B}$, `Cancel` $\mathcal{A} \rightarrow$ `Release` $\mathcal{B}$, and `Cancel` $\mathcal{A} \rightarrow$ `Cancel` $\mathcal{B}$. These dependencies, especially the serial dependency, often contain some *transition condition* that sets a criterium for moving control from one activity to the subsequent one. The transition conditions together with the dependencies form a *rules-based routing* (conditional routing) that enables one or more control flow paths (from multiple possibilities) to be followed (*OR-Split*) depending on the criteria (e.g. a value of variable or a result derived from some complex calculation).

The support for *Parallel routing* is needed to describe concurrently executing activities. In order to achieve parallel routing control flows should be able to split to many parallel threads (*AND-Split*) and later converge into a single thread again. At least two different kinds of converging is needed: *AND-join* for synchronising all the converging threads before control is moved forward and *OR-join* for asynchronous converging. In some cases more advanced rules might be needed for the synchronization (e.g. the control can be moved forward when certain number of converging control flows have been synchronized.)

The workflow process definition should also explicitly specify the *data flow* between the activities.

2.2 The Intelligent Networks case

The term "intelligent networks" (IN) refers to an architectural and functional concept which is intended to ease the introduction of new telecommunication services. Traditionally telecommunication services have been built directly on the hardware of a switch with proprietary tools. This approach is very expensive; it takes several years to specify, develop, test and deploy new services. The crux of the IN approach is that it offers a standardised environment where the software controlling the basic switch functionality is separated from the software which controls the call progression.

The 101-service of Telecom Finland

Telecom Finland is a Finnish network operator and telecommunications service provider with a turnover of ca. FIM 5 billion/year and 6500 employees. Besides the conventional telephone services (local calls, long distance calls, international calls) and telephone services based on the cellular systems technology (GSM, NMT-900, NMT-450), it offers more complex services based on the modern IN switches. Freephone, virtual private network and conference calling are some examples of the services based on the modern switching technology.

The purpose of the 101-service is to make it easy for a customer to direct all his long distance telephone calls to the network of Telecom Finland. In principle this is a very simple service but due to the company organisation and actors beyond the control of the company there are many difficult questions related to the process which realizes the service.

When a customer wants to make a long distance telephone call in Finland, he must either implicitly or explicitly select the operator which arranges the connection from the switch of the local operator to the destination. If the customer does not care which company arranges the long distance connection then he only selects the dialling code and the local telephone number as usual. The operator responsible for the long distance connection is then randomly selected among the operators which provide long distance services in this particular geographical area. If, however, the customer wants to govern the selection of the long distance operator, then he must give an operator code before the dialling. To use a certain long distance operator by default, the customer must sign an agreement with the operator. The 101-service is a way to initiate the service agreement process with Telecom Finland.

An overview of the analysis

Five different actors are involved in the realization of the 101-service. *A customer* is a private person who wants to direct all his long distance calls to the network of Telecom Finland. *A business unit* is the part of the organisation of Telecom Finland which offers the 101-service to customers. There exists exactly one business unit which offers this kind of service to private persons. *An operations centre* manages all the long distance calls related issues of some geographic area. Telecom Finland has several operations centres around the country. *A customer centre* is responsible for both technical and administrative issues related to local calls in a geographic area. While there are several customer centres around Finland, the customer centres and operations centres do not necessarily reside in same places. The services of *a local telephone company* are needed if the customer lives in an area where Telecom Finland does not offer local telephone services.

A customer initiates the service process by making a telephone call. An officer of the business unit receives the order and feeds it into an information system. After the check of the customer's credit information, a written agreement is sent to the customer and the operations centre is requested to establish the connection. When the connection is established the customer centre sends a notification to the business unit which in turn informs the customer about the connection. The long distance telephone calls of the customer are then automatically directed to the network of Telecom Finland.

The main tasks of the regional customer centre is to determine the organisation which is responsible for the local telephone service in the area where the customer lives and to order the physical connection from an appropriate organisation. If Telecom Finland takes care of the local service then the connection order is done via a database. If the local services are provided by another company, a standardised fax message is used to order the new routing of connection.

Several database are used in the production of 101-service. The business unit stores administrative information about the customer in the customer database. The business unit also initially adds the customer to the billing database. The customer centre stores technical user agreement information in the database containing user related technical information. Finally, the operation centre stores records of actual calls in a database which is used as a basis for billing. All the mentioned databases are managed by different database management systems.

Several key observations were made of the environment of 101-service:

- The organisation of Telecom Finland is inherently distributed and heterogeneous. Computers and databases are used extensively.
- The organisation of Telecom Finland must satisfy two kinds of needs:
  - the needs emerging from the customer service and
  - the needs emerging from the management of large and very complex technical systems.

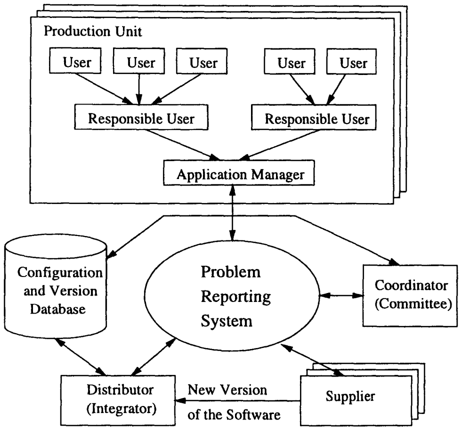

Figure 3 Problem Reporting at Valmet

- The 101-service contains both manual and computerised steps.
- The quality of the 101-service depends very much on how the company manages to co-ordinate and control the various workflows occurring in the service production process.
- There exists actors which can not be steered by Telecom Finland, i.e., the local telephone companies.

The detailed analysis of the 101-service introduced three principal requirements for the workflow tools. First, the formalism which is used to describe the workflow process definitions should allow one to express temporal constraints. Second, there should exist tools which allow the simulation of the process definitions. Simulations should produce information which helps one to predict the time consumed by process instances. Finally, the workflow engine should support the control of process instances by offering the possibility to get snapshots of these instances.

2.3 Problem reporting system of Valmet Corporation

Valmet Corporation is one of the biggest paper mill machinery producers in the world. It has production units in many countries in Europe and in the USA. Valmet has developed a problem reporting system to help corporate wide maintenance of certain central legacy systems of the production units. The problem reporting system helps in getting the faults reported quickly to relevant parties and in having standardised processes for repair. It keeps track of the problem frequencies and of the performance of the software suppliers.

The overall system structure and functions

The problem reporting system embeds a workflow definition between many organizations including production units of Valmet, the corporate level management of Valmet, software suppliers, and a software distributor. Next we outline the interorganizational process definitions. Within each organization there is a local process definition that might be rather complicated. The overall system is depicted in Figure 3, where the rectangles illustrate actors and the arrows data flows.

In the production units each software system has a *responsible user*, who may be able to solve solve some of the problems other users report. Unsolvable problems get forwarded to the *application manager* of the production unit, who can insert a new problem report to the problem reporting system or make searches to find if there are already solutions available.

The problem reporting system is implemented on top of Lotus Notes ((tm) Lotus Development Corp.). Each problem report corresponds to a form, that contains a reporting area for users (application managers), an answering area for suppliers, a common commenting area. Each form contains also the following data:

- Author identity code
- Target software identification (also version number)
- Data access rights for actors
- Status of processing (reported, received, under work, done, distributed, approved, frozen, cancelled)
- Report class (software fault, change request, development inquiry, other report)
- History of status changes
- Performance information

A new report on the help desk first gets the status *reported*. In case of *software faults* and *change requests* the processing includes a series of actions. The main actors here are the reporting application manager, the *supplier* of the software and the *distributor*. The flow is controlled by setting different statuses to the form depending on its processing phase. Other parties can follow how the case develops on the bulletin board. In case of other types of reports, the system serves as a discussion forum.

The report category *software fault* has two subcategories: *fatal* and *operational limitation*. The former subcategory means that the software cannot be used at all, while the latter rather poses constraints on the use of the software. Faults are usually supposed to be repaired in 3 to 5 days depending on the software component. The report category *change request* is used when the coverage by the maintenance agreements is not evident. The report category *development inquiry* contains cases that need cooperative planning. This category contains development initiatives and specification of new versions, and the corresponding processing is cooperative planning with concurrent access to the report by several authors.

The *supplier* has partial access to the corporate level help desk and is responsible for finding from it the relevant reports. After noticing a new *software fault* report (status to *received*) the supplier is supposed to give an initial response containing first hand estimates on the correction time and possible temporary workarounds. This must be done within 24 hours. The supplier is also supposed to provide a correction plan. After the initial response, the status is set to *under work*. After the supplier has shipped the corrected version of the software to the distributor, the supplier changes the status to *done*. The *distributor* gets the fixed version from the supplier and installs and tests it in those production units that want to have it. The distributor maintains and utilises the configuration and version database, that is used to track the corporate wide installations. This database is also used for compatibility considerations. After the distributor has delivered the software to the appropriate production units, the distributor sets the status to *distributed.* The unit that initiated the whole process sets the status to *approved* after it has found the result acceptable.

All areas on a report are readable to all parties except the measuring area which is visible to Valmet personnel only. All parties can write to the common commenting area where each comment is automatically attached with authentication, date and time. Measuring area is visible to Valmet personnel and contains performance indexes to be used to evaluate the behaviour of the suppliers.

On the corporate level there is the *coordinator* (possibly a group of technical managers and consultants) that makes strategic guidelines and decisions on the software architectures and components. The coordinator helps to solve cases where the troubles cannot be clearly localised to a single system. For instance, troubles in interfacing two systems with each other might require solutions at both ends, not only at one end.

Analysis of the system

The underlying bulletin board metaphor provides weak support for workflow thinking. Processing of a workflow is mainly based on the use of the statuses in the reports. It is up to the users to interpret and set the statuses in a proper way. In the future, mechanisms for controlling the status changes should be added. The control conventions are well defined in the handling of software fault and change request reports. Development inquiries, and other reports do not have so clear conventions. In those cases the system functions as a discussion forum and processing becomes cooperative authoring. For instance, the cooperative planning process of a new software version needs support for concurrent access of the shared reports by many actors.

The system involves many autonomous organizations. The approach so-far has been to publicly announce reports to all users. Also competing suppliers can see each others faults. This arrangement works in favour of Valmet. In more general cases of interorganizational workflows data hiding and autonomy considerations ought to be considered. The transaction support within the report database could be made better as the current system maintains only one copy of the data and lets the actors do destructive changes to this data.

The overall response time of the system depends on how active the users are in reading the reports. There is a need to give the system a more active role. For example, it could alarm the responsible actors for delays in processing. This self monitoring could be combined with forecasting capabilities to estimate durations of process instances. In order to give a more active role to the system it might be good to integrate message exchange (e.g. electronic mail) functions to it.

3 SUMMARY OF THE MAJOR SHORTCOMINGS IN THE CURRENT WORKFLOW PRODUCTS

Inadequacy for inter-organisational workflows. Although in some cases high-end workflow products (e.g. FlowMark) might be suited for defining workflows that take place in many organisations, none of the products can deal with wide range of heterogeneity and autonomy issues in in this kind of system. (This is also true in enterprise-wide systems.)

Lacking standards. Although workflow products provide many ways to integrate external applications and documents, they lack standard interfaces that would support interoperability among workflow management systems and third party software. The standards to be developed by the Workflow Management Coalition will help to overcome this problem.

Business process analysis and re-engineering. Workflow products usually do not include tools for BPA and BPR for the basis of the workflow process definition. Moreover, although there is a great variety of separate BPA/BPR-tools the definitions can not be exchanged between them and workflow products.

Inadequate support for testing, debugging and analysis of the scenario definitions. Most of the products enable manual experimenting of the scenario definition before putting it into the production use. However, certain correctness aspects (e.g. the correct termination of the workflow process in all possible cases) can not be tested automatically. Workflow products also usually lack simulation tools.

Missing advanced monitoring tools. While most of the workflow products provide some kind of audit trail and status reporting, only few of them can produce more advanced reports that would e.g. reveal the efficiency of the system and possible bottlenecks in it, thus supporting continuous enhancing of the system.

Inadequate of totally missing definition of the transactional properties. None of the products we know enables the definition of some or all of the traditional ACID properties for an activity or a group of activities.

Moreover, none of the products contains backward recovery by compensation. Neither do current products have a possibility to define contingency activities for alternative execution after a scenario instance is cancelled or cannot be executed for some other reason.

Managing shared resources also has serious shortcomings. In some products some of these shortcomings are solved by preventing the concurrent access of data items and document files related to the certain workflow process instance. Other products leave this problem to the application programmer. Even less is offered by the workflow systems if one needs to control the correctness of the data retrieved from the external data storages that are used concurrently by many workflow process instances (or even by several different workflow management systems). This means that none of the programs we know enable to define all the shared data and its visibility. All this leads also to inadequate support for co-operative workflow applications.

Inadequate definition of inter-activity and inter-process dependencies. Only few workflow products enable the definition of advanced dependencies above the serial dependency (`Release` $\mathcal{A} \rightarrow$ `Begin` $\mathcal{B}$) between the activities while many other types of dependencies might be needed (e.g. `Cancel` $\mathcal{A} \rightarrow$ `Begin` $\mathcal{B}$ and `Release` $\mathcal{A} \rightarrow$ `Cancel` $\mathcal{B}$.

Dynamic and ad hoc features. Typically ad hoc workflows, modifying the rules or data of the scenario instance in the execution and other features empowering the end users are lacking from the workflow systems aimed to production type tasks. However, there are growing number of products including this kinds of features in their functionality.

4 CONCLUSION

In this paper, we have studied three inter-organizational workflow applications, deduced requirements from them and compared the requirements with tools and properties offered by some commercial workflow products. We have concentrated ourselves mainly on aspects like workflow definition tools and advanced transactional properties which seem to be of prime importance in these environments. The comparisons show that the chosen products do not support many features needed in these environments. Especially lacking are the advanced transactional properties and specification facilities which could be used in interorganizational environments.

We have applied the conceptual framework and reference architecture defined by the Workflow Management Coalition. We have found them valuable in assessing the products and applications. Further work needs to be done, however, before they can be used to solve real problems of interoperability.

The results also show that further research is needed to better understand the transactional requirements and their implications in the autonomous workflow environments like those handled here. The process definition tools should support pre-analysis of the transactional properties. What those properties are, is for further study.

Acknowledgements We are grateful to the "TransCoop people" at GMD-IPSI in Darmstadt and at University of Twente in Enschede for clarifying discussions during the application analysis.

We thank PortNet consortium and especially Ms. Ina Mickelsson and Mrs. Eija Berg for mediating information about the system. We also thank Dr. Olli Martikainen, Vice President (R&D) as well as Mr. Panu Hämäläinen and Mr. Atte Nikkanen of Finnish Telecom for releasing information concerning the 101-service and supporting our work. We further thank Mr. Hakan West, Vice President (Systems & Procedures) and Mr. Tuomo Raitanen of Valmet Corporation.

We also thank Mr. Esko Uotila from ICL Personal Systems for kindly providing us the TeamFlow software. We are also grateful to Dr. Frank Leymann and Mr. Harri Puhilas from IBM for interesting discussions and material concerning the FlowMark product. Finally, we thank Mr. Kai Vehmersalo from WorkFlow Oy for information about Staffware product.

REFERENCES

[ASSR93] M. Attie, M. Singh, A. Sheth, and M. Rusinkiewicz. Specifying and enforcing intertask dependencies. In *The 19th International Conference on VLDB*, 1993.

[BDS+93] Y. Breitbart, A. Deacon, H.-J. Schek, A. Sheth, and G. Weikum. Merging application-centric and data-centric approaches to support transaction-oriented multi-system workflows. 22(3), September 1993.

[CR90a] P. Chrysanthis and K Ramamritham. Acta: A framework for specifying and reasoning about transactions structure and behavior. In *ACM SIGMOD International Conference on Management of Data*, 1990.

[CR90b] P. Chrysanthis and K. Ramamrithan. A framework for specifying and reasoning about transaction structure and behavior. In *ACM SIGMOD International Conference on Management of Data*, 1990.

[GH94] D. Georgakopoulos and M. F. Hornick. A framework for enforceable specification of extended transaction models and transactional workflows. *International Journal of Intelligent and Cooperative Information Systems*, sept. 1994.

[GR93] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques.* 1993.

[Gru94] J. Grudin. Computer supported cooperative work: History and focus. *COMPUTER*, May 1994.

[Kle91] J. Klein. Advanced rule driven transaction management. In *IEEE COMPCON*, 1991.

[Loc94] R. Lockwood. Groupware and workflow: the european perspective. In T.E. White and L. Fischer, editors, *New tools for new times: The Workflow Paradigm*, chapter 12, pages 195–214. Future Strategies Inc., Book Division, Alameda, CA, 1994.

[PC94] K. Ramamritham P.K. Chrysantis. Synthesis of extended transaction models using acta. *ACM Transactions on Database system*, 19(3):450–491, sept. 1994.

[SR93] A. Sheth and M. Rusinkiewicz. On transactional workflows. 16(2), June 1993.

[Tra94] TransCoop. Transcoop deliverable 2.1, specification language requirements. Transcoop (ep8012) deliverable, University of Twente, Dec. 1994.

[VEH92] J. Veijalainen, F. Eliassen, and B. Holtkamp. The s-transaction model. In A.K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 12, pages 467–513. 1992.

[Vei90] J. Veijalainen. *Transaction Concepts in Autonomous Database Environments.* R.Oldenbourg Verlag, 1990.

[Vei92] J. Veijalainen. Issues in open edi. In *Proc. of 2nd International Conference on Systems Integration*, pages 401–412. IEEE Computer Society, june 1992.

[Vei93] Jari Veijalainen. Heterogeneous multilevel transaction management with multiple subtransactions. In *Fourth Intl. Conference on Database and Expert Systems Applications, DEXA '93, LNCS Nr. 720*, pages 181–188, September 1993.

[Wäc91] W. Wächter. Contracts: A means for improving reliability in distributed computing. In *IEEE COMPCON*, 1991.

[Wor94] Workflow Management Coalition. *Glossary; A Workflow Management Coalition Specification*, nov. 1994.

[WR92] H. Wächter and A. Reuter. The contract model. In A.K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 7. 1992.

[WS92] G. Weikum and H. Schek. Concepts and applications of multilevel transactions and open nested transactions. In A.K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 13. 1992.

APPENDIX 1 SUMMARY OF THE PRODUCT EVALUATION

Technical details of the products

	Staffware	**FlowMark**	**TeamFlow**
Architecture	Client/Server	Client/Server	Client/Server, Mail-Based
Server platform	UNIX	OS/2, AIX	OS/2, Windows NT
Client platform	UNIX, Windows	OS/2, AIX, Windows	Windows
DBMS	Proprietary, Informix, Oracle	ObjectStore	Proprietary, (SQLServer or SQLBase for audit-trail log)

Workflow process definition functionality

Graphical def.	Yes	Yes	No
Hierarchical def.	No	Yes	No
Reusability	Poor reusability in definition	Every object can be reused	Existing definition as a template for a new one
Language	Script	FDL	No
Process simulation	No	No, only animation	No
Testing before production use	In separate test environment	By animation facility	No separate testing facilities
Import/Export process definitions	Yes	Yes, using FDL	Transferable internal representation
Misc.	Includes form tools		

Workflow process definition features: activities

Activity types	Steps (with a form interface), automatic steps, external events, management reports	Program and process activity	Steps, co-operative and information steps
Compensating or contingency activities	No	No	No

Workflow process definition features: activity assignment

Activity assignment	Single user, group, role	Single user, role, organisational unit	Single user
Dynamic assignment	Yes	Yes	Yes (by user selection)

Workflow process definition features: control flow

Dependencies between activities	Release → Begin, Release → Release	Release → Begin	Release → Begin
Routing	Rules-based	Rules-based	Response-based
Parallel routing	Yes	Yes	Limited
Synchronisation of converging control flows	AND-join	AND-join, OR-join	No
Iteration	Yes	Yes	Yes
Bundles	No	Yes (by using ext. utility)	No
Data flow definition	No	Yes	No

Workflow process definition features: deadlines and other temporal dependencies

Deadlines can be set to	Activities, group of activities and whole process	Activities	Activities
Deadline expression	Duration, absolute, relative (to e.g. some variable) or calculated date and time expressions (years, months, weeks, days, hours, minutes). Deadlines can be conditional.	Duration (days, hours).	Duration (days, hours).
Action if deadline exceeded	Start user defined activities	Notify	Notify
Other temporal dependencies	No	No	No

Workflow process management			
Activity delegation	Yes	Yes (has also automatic delegation to the substitute user)	Yes
Modify a process in execution (by an end-user)	No	No	Yes
Forward recovery	Continue from the last saved state	Continue from the last saved state	Continue from the last saved state
Backward recovery	No	No	No
Work balancing	No	No	No
Data definition features			
Basic data types supported	Text, number, date, time, memo	String, long, float	String
Structured data types	No	Array, structure	List of strings
Misc.	User defined value lists and data tables		
Data management			
Management of the workflow process relevant data	Global within a process instance. Parallel activities can have concurrent access (without any control).	Each activity has private data containers contents of which are exchanged between the activities as defined in data flow definition.	Data are stored in TeamMail mail items. In case of parallel activities, data items are replicated.
Document management	Pointers to physical files using attachment fields	No	Managed as TeamMail attachment files
Concurrency control in the document management	Yes (based on UNIX file handling)	No	Documents are shared as mail attachments
Versioning of the data	No	No	No
User definition			
Single user definition	Predefined set of attributes + user defined attributes	Predefined set of attributes	Predefined set of attributes
Other user structures	Group, Role	Role, organization	No
Integration			
Integration mechanisms	External program calls from Staffware forms. Forms can be replaced by another application. Very limited API support. Visual Basic Form helper utility.	External program (or DLL function) calls, APIs	external program calls, APIs
APIs	Proprietary SAL API, DDE	C/C++, REXX, Visual Basic, DDE, OLE, VIM, HLLAPI	C/C++, Visual Basic
Administration and monitoring functionality			
Audit trail database	Yes	Yes	Yes
Built in audit-trail reporting	Status of process instance, audit-trail report, customised reports	Graphical representation of the process instance status	Status of process instance, audit-trail report, predefined reports with user parameters
Suspend/re-route/re-assign/terminate a process instance	Yes/(Yes)/Yes/Yes	Yes/No/Yes/Yes	Yes/Yes/Yes/Yes
Modify data of a process instance	Yes	Yes	Yes
Suitability for different workflow types			
Workflow type product suits best for	Administrative and production	Production	Administrative and ad hoc