DERI – DIGITAL ENTERPRISE RESEARCH INSTITUTE

# Aspects in Workflow Management

Simeon Petkov     Eyal Oren     Armin Haller

DERI Technical Report 2005-04-10
April 2005

**DERI Galway**
University Road
Galway
IRELAND
www.deri.ie

**DERI Innsbruck**
Technikerstrasse 13
A-6020 Innsbruck
AUSTRIA
www.deri.ie

DERI – DIGITAL ENTERPRISE RESEARCH INSTITUTE

DERI Technical Report
DERI Technical Report 2005-04-10, April 2005

# Aspects in Workflow Management

Simeon Petkov[1] Eyal Oren[1] Armin Haller[1]

**Abstract.**

[1]Digital Enterprise Research Institute, National University of Ireland, Galway.

**Acknowledgements**: This material is based upon works supported by the Science Foundation Ireland under Grant No. 02/CE1/I131.

# Contents

## 1 Introduction

Workflow management deals with supporting business processes in organisations, it involves managing the flow of work through an organisation [2]. Workflows are a collection of coordinated tasks designed to carry out a well-defined complex process [29].

A workflow management system is a generic information system that supports modelling, execution, management and monitoring of workflows. Such a system operates on a workflow specification, a description of the business processes in the organisation that should be supported. A workflow management system can be compared to a database management system: it is a generic system that operates on a schema definition of the (processes in the) organisation.

Many different workflow systems have been developed that focus on different application domains and provide different functionality, c.f. [7, 22, 26]. Workflow management lacks a standardised theory that provides a theoretical background for workflows like the relational algebra provides for databases [2]; despite efforts of standardisation bodies there is no consensus on the representation or conceptual model of workflow processes [34].

In this diverse and complicated landscape it is a challenge to evaluate and compare the functionality of various workflow management systems and to survey the functional requirements of worklow management systems. A number of approaches attempt to address this situation: Jablonski and Bussler [24] describe a number of essential perspectives and aspects of a comprehensive workflow management functionality; they provide a structure in the complex environment of workflows. van der Aalst *et al.* [7], Russell *et al.* [32] systematically analyse available functionality in existing workflow management systems, and categorise these in a number of *workflow patterns*; these patterns are devoid of implementational issues and form a qualitative standard against which existing workflow management systems can be benchmarked.

Our work integrates and extends these approaches; we describe various aspects of workflow management; we indicate how these can be modelled and what quality indicators can be used to assess their support in workflow management systems. This overview is a preliminary to a theoretical and practical survey of modelling formalisms and workflow management systems. These surveys are performed in the *m3pe* project, and are available on `http://www.m3pe.org`.

## 2 What is workflow management

Different interpretations of workflow management can be found in the literature. The Workflow Management Coalition is a standardisation body that tries to define some common grounds in the field and establish standards for workflow interoperability. The Workflow Management Coalition has published a term glossary, defining common workflow terms [35].

A workflow can be defined as “the computerised facilitation or automation of a business process, in whole or part”. When employing workflow technology the *what is intended to happen* is defined in a *business process*. This represents the real world sequence of activities usually happening within an organisation. The predefined plan telling us *how* such a business process has to be carried out is to be found in a so-called *process definition*. Our plan defined in the process definition enumerates the *steps* to be taken; it is composed of single *manual activities* (such that have to be carried out manually) or *automated activities* (such that can be carried out by the system).

Having a process definition defining automated steps we can use a *machine* to control them. This system is called a *workflow management system (workflow management system)*. What is actually

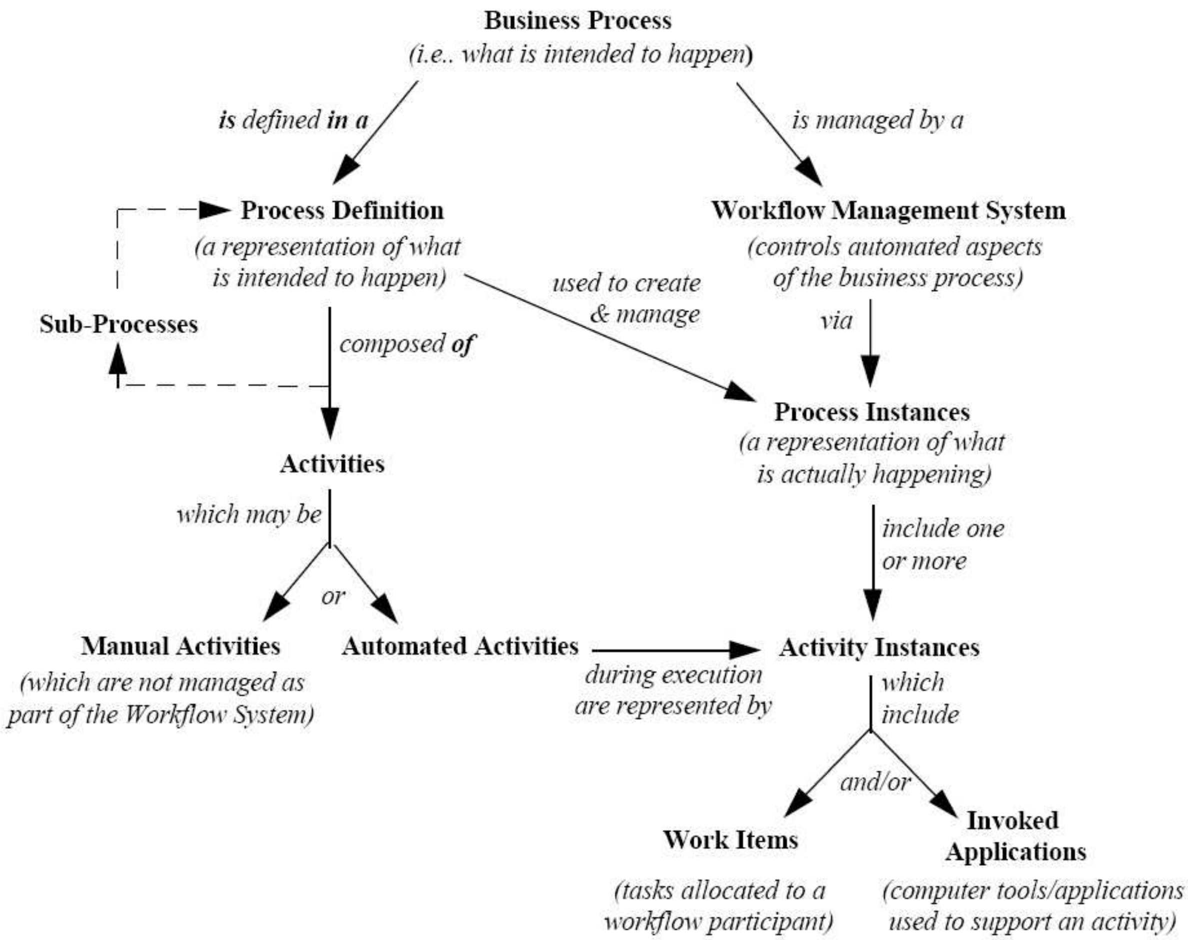


Figure 1: Terms in Workflow Management, from [35]

happening in the workflow management system when executing a process definition is represented through a *process instance*. The process instances are composed of the *activity instances* according to their definition. If an activity instance is the instance of a manual activity its execution results in a *work item* which is to be carried out by *workflow participant*, and in case that it is the instance of an automated activity it results in an application being invoked by the workflow management system (*invoked application*).

**Workflow management systems** A workflow management system is a software system that manages workflows; it consists of several elements [13]. It operates on a workflow *representation*, a description of the business process. The *execution semantics* of the system defines the dynamic behaviour of workflow instances: the execution semantics gives meaning to the syntactic representations of workflows, it prescribes how the system behaves in the presence of a workflow description. The workflow management system has a certain *architecture* that describes its construction, its components, and their communication and interaction patterns. The *implementation* of the work-

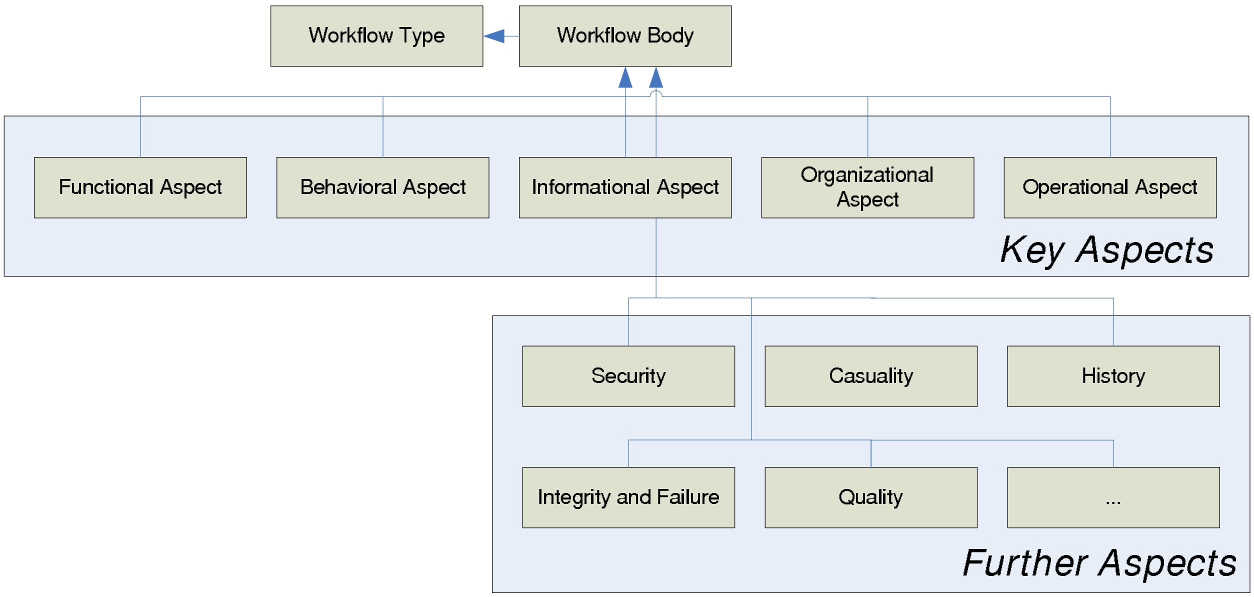


Figure 2: Aspects in workflow management

flow management system is the concrete software system that executes workflow instances based on the workflow definitions and the defined execution semantics.

## 3 Aspects in Workflow

In the literature different opinions can be found on the aspects of workflow management. Opinions differ on whether these should be called *aspects* or *perspectives*[1], on which aspects should be distinguished, on how to call those aspects, on which aspects are most relevant, on what comprises a certain aspect, and on how to model or quantify the elements of a certain aspect. An aspect is a set of elements in a workflow metamodel that target some subset of self-contained functionality (e.g. data management or organisational management).

In the following section we give an overview of those opinions different people distinguish and categorise this overview per aspect. We investigate for each aspect the prevailing opinions on (i) what comprises this aspect, (ii) how important is this aspect in workflow management, (iii) how to model the elements of this aspect and (iv) what quality indicators exist for this aspect.

The composition of the following chapter is twofold: firstly we introduce the *key-aspects* identified by [24]. These are the five aspects in workflow which are necessary in order to provide comprehensive workflow functionality and are widely accepted as such by the workflow technology area. Following we identify possible additional aspects and discuss extensibility characteristics of workflow meta-models which come to mind given the fact that additional aspects can occur. However, alternative categorisations exist and alternative approaches give more importance to different aspects, c.f. [12, 25, 31, 33]. Figure 2 shows the relations between aspects in workflows.

[1]We consider a perspective a subdivision of an aspect, i.e. one aspect can be composed of multiple perspectives.

### 3.1 Functional aspect

**What comprises this aspect?** The functional aspect defines *what* has to be done. It contains the definition of activities in the workflow, and an hierarchy of workflows and sub-workflows.

**How to model this aspect?** The modelling characteristics of a workflow is twofold:

- composition: we distinguish *simple* and *composite* workflows. A *simple workflow* does not include definitions of sub-workflows; a *composite workflow* contains one or more sub-workflows.
- constraints: the definition of a workflow allows the assignment of constraints. A *workflow constraint* is a set of rules which allow to perform consistency-checks upon the definition and the execution of a workflow. Three types of constraints can be distinguished. *Enter constraints* are evaluated when a workflow goes into production. *Exit constraints* check whether a workflow has completed normally, and are checked at a transition. *Run time constraints* ensure continuous consistency during run time; the implementation of "continuous“ can be done in different ways (e.g. every time a sub-workflow completes, before and after an operation of a workflow is completed, or whenever a data change occurs).

**How important is this aspect?** This aspect provides a frame for the embedding of all the others. It is a prerequisite for the correct functionality of the others.

**What quality indicators exist?** We did not find any metrics for the measurement of the quality of the functional aspect in literature. We identify the following two points as prerequisites for comprehensive workflow functionality:

- composition: the flexibility provided for the definition of workflows is one metric for the quality of this aspect. It is to be distinguished whether the definition of *sub-workflows* is possbile, whether defined workflows can be *reused* in later definitions of other workflows, and whether *dependencies* between sub-workflows can be defined.
- constraints: the definition of a workflow is one of the spots where constraints for the execution can be provided. Constraints defined on the workflow-level are common for all operations defined in the operational aspect of the body.

### 3.2 Behavioural Aspect

**What comprises this aspect?** The *control flow* or *process* perspective describes the activities and their execution ordering. To achieve this one can use different constructors that permit flow of execution control. Activities in this perspective are either atomic units of work or compound activities that consists of an execution order of a set of activities.

**How important is this aspect?** According to van der Aalst *et al.* [7] this perspective is essential in a workflow specification and the other perspectives are built around it: the data flow rests on the control flow, while the organisational and operational perspective are subordinate to it; these other perspectives are defined on top of the control flow and cannot exist without it.

As another measurement of its importance, [3] states that for verification of a workflow specification only the control flow is relevant. The argument is that those perspectives lie mostly outside the control of the workflow management system: the production data is updated by external applications (not by the workflow management system), the execution of external applications is only initiated but not controlled by the workflow management system, etc.

The point being made is that the control flow perspective is the only one which is truly under control of the workflow management system; this is reasonable since it represents the core functionality of a workflow management system: to decide who should do what when.

**How to model this aspect?** Different formalisms can be distinguished that are used to model the behavioural perspective, ranging from informal textual descriptions or semi-formal UML diagrams to formal models like Petri nets or statecharts.

Georgakopoulos *et al.* [21] divide process modelling in workflow management into two basic categories: activity-based and communication-based. The communication-based approach focuses on the communication between people and the commitments that follow from it; the activity-based approach focuses on the activities people engage in and the results of those activities. The approaches are not mutually exclusive but describe the focus of the business modeler when constructing a workflow model. In the communication-based approach one also models activities (because to fulfil their commitments people have to engage in activities) but the focus lies on understanding why people engage in their activities.

Although the communication-based approaches have not found widespread use[2], the approach is worth mentioning. It is based on the Language/Action paradigm which states that communication is used not only to convey information but also to coordinate actions (cf. [9, 10, 17]). The presumption of communication-based workflow modelling is that by observing and modelling the communication and coordination between people in an organisation we will find a better picture of the organisation and its processes (cf. [16]). Since workflow management is about coordinating the work to be done, a model of the coordination (i.e. communication) processes in the organisation is a natural starting point.

Petri nets are advocated as control flow formalism by a number of researchers (see [2, 24]). Some reasons include that Petri nets have a formal semantics, that they are state-based and that they are well analysable [1].

Eshuis and Wieringa [20] propose UML activity diagrams as modelling technique with a specific reactive semantics. These diagrams would be better suited for modelling open and reactive systems (which workflow management systems are) than Petri nets. The afore-mentioned advantages of Petri nets are, according to them, not specific to workflow modelling and, since Petri nets were invented before workflow modelling, their semantics is not specifically intended for workflow modelling.

**What quality indicators exist?** van der Aalst *et al.* [7] have developed a number of *workflow patterns* that provide the basis for an in-depth comparison of a number of workflow management systems. A workflow pattern is defined as the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts. These patterns identify a comprehensive workflow

[2] for instance, of all systems compared in [7], ActionWorkflow is the only communication-based one.

functionality[3] and provide the basis for comparing workflow management systems and evaluating the suitability of workflow languages.

In a number of publications the workflow patterns are presented as an evaluation benchmark for functionaly of workflow management systems; various workflow management systems, modelling languages and web service composition languages are evaluated against them, see [4, 5, 6, 7].

### 3.3 Informational Aspect

**What comprises this aspect?** According to Jablonski and Bussler [24] the informational aspect is defined by the *data* and *data flow* perspective. The main task is to provide the right data at the right time, whereas documents are not a modeling construct in the workflow model per se, they are modelled as data in general. One can distinguish *production data* and *control data*: production data is is produced and used in the business process, control data is used to manage that process. Comparing this definition with the Workflow Management Coalition's *data structures* [35], this corresponds to *workflow control data*; the Workflow Management Coalition's definition separates production data in *workflow relevant data* and *application data*.

Production data is external data (which exist even without workflow) and might be used for it (e.g. an insurance claim). Operations on this data are not provided by the workflow management system itself, but by data integration with external systems. This can for example be achieved by connecting the workflow system to databases used by external application systems. If the workflow system acts as an *enterprise application integration hub*, a conversion of the external data might be necessary. As mentioned above the Workflow Management Coalition further classifies workflow relevant production data. This is external data which is used by the system to determine the state transitions of a workflow instance, for example within pre- and post-conditions, transition conditions or workflow participant assignment. Hence this data can be manipulated by the workflow applications as well as by the workflow engine.

Control data only exists for the workflow management; they are relevant only internally for a workflow management system (e.g. local variables or workflows). When the workflow instance is discarded the relevant control data is also discarded. It represents the dynamic state of the workflow system and its process instances. Control data includes for example state information about workflow instances, or state information about activity instances.

**Data** The three basic modelling elements of the informational aspect are parameters, variables and the data flow itself. Parameters and variables are typed, which means the structure of the data is implied by its type (typically a workflow management system will understand the structure of such data and may be able to process it).

Data can be represented in different ways, but in current workflow management systems we can make a common object-oriented differentiation of data types in primitive types like byte, char, short, boolean, int, long, float, double which are supported by most of the tools [32] and composite types like array, set or map. A further distinction can be made between the value of data and the pointer to data. Furthermore constraints can be attached to data types like that an integer value

[3] note that these patterns do not describe any minimal business requirements, they are a systematic summary of available features in existing systems. An open question is which of these patterns are essential and necessary for a usable workflow management system.

can only be even. Like constraints an initialisation value can be specified for data which defines the value in the point it is created.

Parameters are used to pass data between workflow types or workflow operations. According to Jablonski and Bussler [24] parameters can have three different kind of values: IN, OUT and INOUT, which relates to their attribute type, if they are read or write-only or read and write arbitrarily. Parameters are scoped to their workflow type or operation and can be defined mandatory or optional. Like with the data, parameters can be either value or reference parameters.

Local variables are typed and available within the respective workflow itself. They are generally user defined, if they are system defined they are called workflow context variables.

**Data Flow** The actual data flow happens between parameters and workflow local variables, whereas the flow follows a source target relationship. It can be direct from OUT parameter to IN or from OUT to a local variable. In order to avoid syntactical or semantical mismatches between different data types, conversion functions are necessary. This means the flow of data can be first through a conversion function until it reaches the IN parameter of an operation. Access to production data which is (as defined above) not internal to the workflow system can be achieved by conversion functions or within workflow operations. If it happens over conversion functions it remains automated and a user interaction can not be performed. Hence it is more appropriate to integrate access to external data into workflow operations.

**How important is this aspect?** While the heart of a workflow management system lies in the definition of control flow process models, operations on data are the main contribution of the system. There even exist meta models which see the data flow as the main aspect [14]: the *data-flow paradigm* views a workflow as a repository for data that is passed between processing components according to sets of rules, current state, and historical data [11]. The data-flow paradigm is well-suited for domains where the workflows tend to be partially specified, dynamic, and goal oriented.

**How to model this aspect?** To evaluate different modelling approaches we first define requirements imposed on the formalism to sufficiently represent the informational aspect in workflow models. First a workflow meta-model should support the specification of the data consumed and produced by workflows. Secondly, data flows among workflows should be explicitly definable. If workflows only exchange data implicitly (e.g. via a commonly accessed database) the reuse of existing workflow types is complicated.

Thirdly an independent definition of the data flow from the behavioural aspect is advantageous in certain cases. As an example, consider a workflow with two subworkflows where the exact execution order of the two subworkflows is not known before execution time, but each of them uses data produced by the other one if available. Here it would be beneficial if only the data flow from one subworkflow to the other one could be defined without specifying any control flow.

Since data type declarations follow the same schema in most languages it is of no relevance what formalism to use to model data in workflow systems. As graphical means for characterising the information contained in the data perspective one can use ER diagrams, class diagrams and object-role models. Current workflow management systems are not widely utilising these kind of modelling technique. Current tools focus largely on the definition of process structures and their associated components in a graphical form. The support for the data perspective is typically fragmented and distributed across various parts of the overall workflow model [32].

**What quality indicators exist?** Russell *et al.* [32] introduce *data patterns*, which aim to capture the various ways in which data is represented and utilised in workflows. They identify 39 data patterns recurring in workflow management systems divided into four distinct groups: data visibility patterns, data interaction patterns, data transfer patterns, and data-based routing patterns. These patterns are abstractly specified, independent of a specific workflow technology or modeling language. Doing so they can be used as a benchmark for the evaluation of the data-aspect of workflow management systems and models.

### 3.4 Operational Aspect

**What comprises this aspect?** According to Jablonski and Bussler [24] the operational aspect is defined by the workflow application perspective. Workflow operations are implemented by application programs. These can be very different in nature but from the workflow point of view the technical details of the application programs are to be kept transparent. Workflow applications are application programs which integrate transparently into the workflow management system and by doing so allow this kind of abstraction. This is achieved by the operational aspect by providing an interface which decouples the workflow management system from the actual implementation of an operation.

**How important is this aspect?** Operations are meant to integrate transparently into the workflow management system. Therefore from the point of view of the modeler the operational aspect is seen as nothing more than an interface. There is a minimum set of requirements which are to provided by this interface (discussed in the following paragraph) but the details of the implementation of this aspect are not of importance to the user. However for the workflow management system they could be interesting for performance issues.

**How to model this aspect?** The following composition of the introduced interface is proposed:

- *Parameters.* The parameters required by the application have to be provided. The definition of them is done in the definition of the operation and the the presence of the according values is handled by the information perspective (Section 3.3).

- *Interaction Mode.* An interaction mode is required to indicate whether user interaction is required.

- *Coupling.* It has to be specified if the invoked program has to be executed within the sphere of control of the workflow management system.

- *Invocation Mode.* For efficiency reasons the workflow management system has to know wether to execute an application synchronously or asynchronously. However for the end user this fact is not of interest.

- *Settings.* If settings might change when executing an application several times this should be made visible e.g. through an appropriate parameter. This can be provided ad hoc if it's value is only to be determined at runtime or can be hard-coded in the definition of the interface if it is static.

There is also information flowing back from the external application to the workflow management system:

- *Actions.* If an application allows the user to influence the execution, a list of actions can be made available.
- *Interaction Mode, Coupling, Invocation Mode and Execution Location.* This is provided to the workflow management system for status reporting.
- *Error Code.* This way the workflow management system is notified about the success of the execution of an operation.

**What quality indicators exist?** Comparable to patterns for the behavioural and informational aspects, a set of *interaction patterns* (following the approach from [32] and [7]) is planned to appear in the near future.

### 3.5 Organizational Aspect

**What comprises this aspect?** The organisation perspective defines *who* is responsible for doing the work in a workflow. Or in other words: this perspective allows to assign constraints to activities which determine requirements on the agent (be it human or non-human) responsible for carrying it out in the context of the organisational framework the workflow management system is acting in.

In different literature (as well as in the implementation of different workflow management system) opinions about the scope of the organisational aspect in workflow management differ dramatically. Whereas some systems and publications ignore this aspect as a component of workflow management others discuss it in detail and state complex requirements on it. Bussler [13] proposes the following criteria:

- Representation of the organisational structure:
  - Scope: Global vs. local. An organisational structure can be global for all workflows. Evaluation of assignment rules in all workflows are modelled upon the same organisational structure. Another option is to construct a single (private) organisational structure for each workflow type. So to speak a private view on the organisation. Finally a mixed approach can be taken, where some parts of the structure are global and others local.
  - Type: What is being modelled? Systems can be categorised upon the coverage and the expressive power of the organisational schema. E.g. if only individual task holders, roles or even more complex modelling elements such as groups or places are present. Furthermore here we can distinguish systems with an extensible schema and analyse the schema on consistency rules or derivations. Finally the access abstraction is analysed: is the data being accessed directly or are there any access-functions present.
  - Dynamics: This criterion divides the systems which allow changes during runtime and such which do not. Hereby the versioning comes up - the possibility to make the mentioned changes in a controlled manner is an important point.

- Autonomy and Openness: Access control and authorisation. Since the access on organisational elements should happen in a controlled way systems which allow the authorisation of the access and have the facility to protect single objects are being distinguished. A further criterion is the access on objects by programs.

- The handling of work assignment:

  - Type: Roles vs. complex rules. Systems can be categorised by the type of assignment. E.g. it can be distinguished if the assignment is made on a role-basis or if the system allows more complex assignment rules.
  - Context: Which data is accessible? This criterion categories systems based upon the data which is available during the assignment procedure (Workflow Parameters, History Information, Workflow States, etc.).
  - Type of Rules: Fixed set of rules vs. extensible/combinable rules? Whereas some systems only provide a fixed set of rules which can be used as it is, other allow the combination and extension of the rule set.
  - Scope: Scope of the rule: a rule can apply for one single activity, one sub-workflow, one workflow, multiple workflows or to a combination of these.
  - Rule Evaluation: When? This criterion categories systems upon the moment when rules are being evaluated. E.g. they can be evaluated as early / as late as possible, complete or partially, once or multiple times and they can be re-evaluated upon a change on the underlying data.
  - Subject. In some systems a workflow is being assigned to an agent, who/wich then has the authority to execute all activities/operations in this workflow; in other ones the assignment is per-activity/-operation.
  - Result: Role vs. instance. The result of the evaluation can be a role vs. a concrete instance from the data set.
  - Autonomy of Rules: Are the rules protected for modification in some way? As already mentioned the assignment rules have to be protected in some way. Therefore systems which allow protection of the rules and such which do not are being distinguished.
  - Dynamics. Based upon this criterion systems which allow a dynamic change of the assignment rules are being distinguished. This can happen e.g. through re-evaluation.
  - Pragmatics: “Must” execute vs. “can” execute. This criterion divides the systems in such which assign work to an agent and such which the agent choose himself.

- Synchronisation of work assignment:

  According to Jablonski and Bussler [24] an organisational model in the context of workflow management consists of an organisational structure and an organisational population. The organisational structure defines the objects in the organisation (like “user”, “role”, “group”, etc.) as well as the relationships between them (like “supervises”, “is member of”, etc.) The organisational population instantiates the so defined structure by i.e. specifying concrete users and their roles.

> Having such a representation of the organisational structure and population workflows have to be assigned to specific users. Normally more than one user is eligible to execute a task. The authors describe the complexity of this assignment as twofold: (i) if one user starts performing a task others have to be stopped from performing it. This implies two different synchronisation scenarios, namely a "1-out-of-many" rule (in the case of one single user being notified about the task and performing it) or a "*-out-of-many" rule (if every eligible user can carry out the task and all of them are being notified). (ii) Since not everybody is eligible to execute all tasks rules have to be in place to specify which users are eligible to execute specific tasks. These are called Task Assignment Strategies or Organisational Policies.
>
> Dustdar [19] and Bernauer *et al.* [12] tackle the need for support for ad hoc and collaborative processes in virtual teams for inter-organisational workflows and provide a flexible synchronisation approach for the assignment of activities.
>
> Since the synchronisation of work is a common topic, literature provides a common state-transition diagram handling the assignment of work items. This particular approach stands out by providing two additional states - namely: "delegated" and "forwarded". In an inter-organisational workflow these become necessary since an item forwarded to a business partner has to be distinguished in some way.
>
> One open point remains: the synchronisation of assignment of work to multiple participants (e.g. to a team). We have not found an adequate solution throughout this survey by now.

Zur Muehlen [27] uses the organisational aspect of workflow models as a reference for the comparison of workflow management systems. The author claims that in order to be able to represent a wide spectrum of organisational structures an object oriented concept should be used as a foundation for the representation. The classical organisational elements (like organisational unit, roles, etc.) should be provided but the user should be able to add own constructs using the underlying concept. In [28] assignment policies, synchronisation policies, resource integration and run time issues of the organisational aspect are discussed.

Workflow management systems should be robust against changes in the building organisation. If employees leave, change their position within the organisation or are hired this should not have an impact on the workflow model. This can be achieved through a so-called dynamical role resolution whereby the responsible actor for carrying out a specific task is being determined at runtime. For addressing a responsible person the workflow management system should be able to provide specific requirements on the actor on the one hand and should also provide the facility of assigning a task to multiple actors (e.g. to an organisational unit) on the other hand. The execution of workflows should also be possible when a designated actor is not present. Therefore the system should provide appropriate substitution mechanisms.

Curtis *et al.* [15] name the following as an "essential basis for process modelling abstracted from relevant literature": an *agent* is an actor (human or machine) that performs a process element; a *role* is a coherent set of process elements that can be assigned to an agent as a unit of functional responsibility, and an *artifact* is a product created or modified by the enactment of a process element.

**How important is this aspect?** Different than in the behavioural and informational aspect, the importance of the organisational aspect in a workflow management system depends on the application domain. It is for example hardly relevant in web-service composition languages (or at least not addressed by standards like WS-BPEL [8]). In workflow management systems that distribute work among human agents opinions differ wether the organisational aspect belongs within the workflow management system or in an an external component [18, 36]. Some of them stress the organisational aspect strongly and provide sophisticated modeling constructs for the organisation and policies [13], others provide minimal support or do not consider the aspect at all [8, 36].

**How to model this aspect?** The organisational model is an abstract representation of the organisational hierarchy. In order to represent this hierarchy an organisational meta-model has to be provided, c.f. [28]. In order to ensure that the system is capable of covering the structure of arbitrary organisations this model has to be extensible. I.e. in addition to the modelling elements of the above meta-model the user has to be granted access to the meta-meta-model elements as well. In this case the possibility of creating / deriving new classes should be present.

The presence of the organisational model enables the assignment of work to "roles", i.e. elements of the organisational hierarchy. Constraints on the organisational structure enable the selection of eligible agents for the execution of work. Depending on the system these are represented in some logic-based language (SQL [23], OCL [30]).

**What quality indicators exist?** We are unaware of an explicit measure that can be used for benchmarking of the organisational aspect.

### 3.6 Further Aspects

As we described opinions on what is important in workflow differ. Comparing various workflow management systems, it is clear that some differ by their nature and some differ in the specific problem areas that they tackle (e.g. office automation and web service composition).

Therefore the aspects present in different technologies and the expressive power of each aspect within one such can vary dramatically. Depending on the specific applications one can choose to include alternates. Further aspects including *security, causality, history, integrity and failure recovery, quality and autonomy*; more advanced scenarios can be found in [24].

In this work we concentrate on the five so called *key aspects* which are common for most allround-workflow management systems and are not specific for single problem-domains. For alternate views on aspect emphasis in workflow technology see Bernauer *et al.* [12], Reijers [31], Sadiq and Orlowska [33].

## 4 Comparison and Conclusions

We provided an abstract introduction to workflow management, and discussed the evolution of different workflow models. We described the five most relevant workflow aspects: the functional, behavioural, informational, organisational, and operational aspect. For each aspect we surveyed how it can be modelled and what quality indicators can be used to measure its support in modelling techniques and workflow management systems. This survey has been performed as part of the *m3pe* project; we will use it in our evaluation of existing workflow management systems.

## References

[1] W. M. P. van der Aalst. Three good reasons for using a Petri-net-based workflow management system. In S. Navathe and T. Wakayama, (eds.) *Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96)*, pp. 179–201. 1996.

[2] W. M. P. van der Aalst. The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[3] W. M. P. van der Aalst. Workflow verification: Finding control-flow errors using Petri-net-based techniques. In W. M. P. van der Aalst, J. Desel, and A. Oberweis, (eds.) *Business Process Management: Models, Techniques, and Empirical Studies*. 2000.

[4] W. M. P. van der Aalst and A. H. M. ter Hofstede. Workflow patterns: On the expressive power of (Petri-net-based) workflow languages. In K. Jensen, (ed.) *Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002)*. 2002.

[5] W. M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: Yet another workflow language. *Information Systems*, 30(4):245–275, 2005.

[6] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Advanced workflow patterns. In *Proceedings of the 7th International Conference on Cooperative Information Systems (CoopIS 2000)*, vol. 1901 of *Lecture Notes in Computer Science*, pp. 18–29. 2000.

[7] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.

[8] T. Andrews *et al.* Business process execution language for web services, version 1.1. Tech. rep., OASIS, 2003.

[9] E. Auramäki and K. Lyytinen. On the success of speech acts and negotiating commitments. In E. Verharen, N. van der Rijst, and J. L. G. Dietz, (eds.) *Proceedings of the First International Workshop on Communication Modeling, the Language/ Action Perspective (LAP'96)*. Oisterwijk, 1996.

[10] J. Austin. *How to Do Things with Words*. Claredon Press, London, 1962.

[11] D. Barbara, S. Mehrotra, and M. Rusinkiewicz. INCAs: Managing dynamic workflows in distributed environments. *Journal of Database Management, Special Issue on Multidatabases*, 7(1):5–15, 1996.

[12] M. Bernauer, G. Kappel, G. Kramler, and W. Retschitzegger. Specification of interorganizational workflows - A comparison of approaches. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, pp. 30–36. 2003.

[13] C. Bussler. *Organisationsverwaltung in Workflow-Management-Systemen*. Ph.D. thesis, University of Erlangen, 1997.

[14] A. Cichocki, A. S. Helal, M. Rusinkiewicz, and D. Woelk. *Workflow and Process Automation: Concepts and Technology*. Kluwer Academic Publishers, 1998.

[15] B. Curtis, M. I. Kellner, and J. Over. Process modeling. *Communications of the ACM*, 35(9):75–90, 1992.

[16] J. L. G. Dietz. The atoms, molecules and fibers of organizations. *Data & Knowledge Engineering*, 47(3):301–325, 2003.

[17] F. Dignum, H. Weigand, and E. Verharen. Preface. In F. Dignum and J. L. G. Dietz, (eds.) *Proceedings of the Second International Workshop on Communication Modeling, the Language/Action Perspective (LAP'97)*. Veldhoven, 1997.

[18] W. Du, J. Davis, Y.-N. Huang, and M.-C. Shan. Enterprise workflow resource management. Tech. Rep. HPL-1999-8, Software Technology Laboratory, HP Laboratories Palo Alto, 1999.

[19] S. Dustdar. Caramba: A process-aware collaboration system supporting ad hoc and collaborative processes in virtual teams. *Distributed and Parallel Databases*, 15:45–66, 2004.

[20] R. Eshuis and R. Wieringa. Comparing Petri nets and activity diagram variants for workflow modelling-A quest for reactive Petri nets. In H. Ehrig, W. Reisig, and G. Rozenberg, (eds.) *Petri Net Technologies for Communication Based Systems*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

[21] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workflow management: from process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, 1995.

[22] L. J. Hommes. *The Evaluation of Business Process Modeling Techniques*. Ph.D. thesis, Delft University of Technology, 2004.

[23] ISO. Database language SQL. 1992.

[24] S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture and Implementation*. International Thomson Computer Press, 1996.

[25] Y. Lei and M. P. Singh. A comparison of workflow metamodels. In *Proceedings of the ER-97 Workshop on Behavioral Modeling and Design Transformations: Issues and Opportunities in Conceptual Modeling*. 1997.

[26] R. Marschak. Perspectives on workflow. In *New Tools for New Times: The Workflow Paradigm*. Future Strategies Inc., California, USA, 1994.

[27] M. zur Muehlen. *Der Lösungsbeitrag von Metamodellen und Kontrollflußprimitiven beim Vergleich von Workflowmanagementsystemen*. Master's thesis, Westfälische Wilhelms-Universität Münster, 1996.

[28] M. zur Muehlen. Organizational management in workflow applications – issues and perspectives. *Information Technology and Management*, 5:271–291, 2004.

[29] S. Mukherjee, *et al.* Logic-based approaches to workflow modeling and verification. In J. Chomicki, R. van der Meyden, and G. Saake, (eds.) *Logics for Emerging Applications of Databases*, chap. 5, pp. 167–202. Springer-Verlag, Berlin, 2004.

[30] OMG. UML 2.0 OCL specification. 2003.

[31] H. A. Reijers. *Design and Control of Workflow Processes*. Ph.D. thesis, Technische Universiteit Eindhoven, 2002.

[32] N. Russell, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst. Workflow data patterns. Tech. Rep. FIT-TR-2004-01, Queensland University of Technology, Brisbane, 2004.

[33] W. Sadiq and M. E. Orlowska. On capturing process requirements of workflow based business information systems. In *Proceedings of the 3rd International Conference on Business Information Systems (BIS99)*. 1999.

[34] A. P. Sheth, W. M. P. van der Aalst, and I. B. Arpinar. Processes driving the networked econony. *IEEE Concurrency*, 7(3):18–31, 1999.

[35] The Workflow Management Coalition. Terminology and glossary. 1999.

[36] The Workflow Management Coalition. The workflow reference model. 1999.