Rosemann, Michael; zur Muehlen, Michael: Evaluation of Workflow Management Systems - a Meta Model Approach. In: Siau, K.; Wand, Y.; Parsons, J. (Eds.): Proceedings of the 2nd CAiSE/IFIP 8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD `97), Barcelona June 16th-17th 1997.

Evaluation of Workflow Management Systems - A Meta Model Approach

Michael Rosemann, Michael zur Muehlen
Department of Information Systems
Westfaelische Wilhelms-Universitaet Muenster
Grevener Str. 91
48159 Muenster
Germany
{ismiro, ismizu}@wi.uni-muenster.de
Phone +49-251-9275-104
Fax +49-251-9275-119

Abstract

The automated enactment of processes through the use of workflow management systems enables the transfer of control flow from application systems to workflow management systems. By now a large number of different systems, that follow different interpretations of the workflow paradigm, are available. This leads to the problem of selecting the appropriate workflow management system for a given situation. In this paper we outline a meta model approach for the evaluation of different workflow management systems. After a general introduction on the topic of meta modeling the meta models of the workflow management systems WorkParty (Siemens Nixdorf) and FlowMark (IBM) are compared as an example.

1 Basics of workflow management systems

Similar to the transfer of data management away from application systems toward database management systems the transfer of control flow enabled by workflow management systems has a profound meaning for the future architecture of information systems.[1] JOOSTEN defines a *workflow management system* (with reference to the workflow management coalition)[2] as „[...] a system that defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the business process logic.“[3]

Workflows[4] are processes - temporal and logical sequences of functions that are necessary to perform operations on economically relevant objects - with automated transitions, namely processes, whose control logic lies within the control of an information system. Every workflow is based on a process model that has been enhanced with additional attributes that allow its automation, the so called workflow model. Workflow models are usually described using directed graphs whose knots represent (elementary or composite) functions. From the point of view of the workflow management system elementary functions are encapsulated, i.e. the workflow management system has an external view on the automated or manual execution of the functions but is not concerned with the execution of the functions themselves.

By now a large number of commercial workflow management systems are available. These systems differ with regard to their efficiency due to their distinct historical background.[5] The selection of an appropriate workflow management system demands an efficient selection process. During the traditional

---

1 Cf. Bartholomew (1995).

2 Cf. WfMC (1996).

3 Joosten (1995), p. 4.

4 For a variety of interpretations of the term workflow cf. e.g. Georgakopoulos, Hornick, Sheth (1995), pp. 122f.

5 Cf. Joosten (1996), p. 2: “The [...] field of workflow management suffers from confusion caused by weakly defined concepts and a lack of consensus about the way in which these concepts are used.”

process of software evaluation the main focus lies on technical aspects, such as reliability, usability, maintainability and adaptability[6] as well as on economic aspects of the systems analyzed. The use of criteria catalogues represents the common practice of software evaluation. However, they fail during the evaluation of modeling methods, because they do not reflect the vast number of modeling alternatives offered by a method. The complexity of these alternatives can be reduced by the formalization of the method's description.

In the next sections we will analyze how meta models can help selecting the appropriate workflow management system. To begin with, we discuss meta models in general and meta data models in particular (chapter 2). In the following chapter the product individual meta data models of WorkParty (SNI) and FlowMark (IBM) are presented. The comparison of the two meta data models allows us to draw some substantial conclusions (chapter 4). The discussion of this approach is the subject of the last chapter.

2 Definition and intention of meta models

2.1 Characteristics and occurrences of meta models

A model is an immaterial representation of a part of the real world. It is created for the purpose of a subject[7], relates two systems and therefore consists of three components: [8]

- The *object system* represents the subjective interpretation of a selected part of the real world (world of discourse) including the relevant part of the environment.
- The *model system* represents the subjective image of the object system. A syntax (also: notation, language) is needed to create the model system.
- The *projection* formulates the relationship between the object system and the model system. The complexity of the real world, that is generated by the number of real world elements and relationships, is reduced by eliminating irrelevant elements. The variety of real world elements and relationships is controlled by clustering elements and relationships (type generation).

If a model system M1 represents the object system of a model system M2, then the model system M2 represents the meta model system of the object system M1 is based upon.[9] Because of this degree of abstraction a *meta model* can be seen as „a design framework, that describes the basic model elements and the relationships between the model elements as well as their semantic. This framework also defines rules for the use and specialization of model elements and relationships." [10]

In the context of information modeling we distinguish meta models describing a notation and meta models describing a procedure.[11] *Meta data models* characterize notations that can be used for information modeling purposes. *Meta process models* describe the modeling process using a specific method. Every meta model is based upon another meta model, which can be of the same kind, e.g. the notation of the entity relation diagram may be explained using another entity relation diagram. If the similarities of a number of meta models are consolidated in one universal model that also claims a high degree of semantic quality we speak of a *reference meta model*.

6 Cf. Dunn (1993); Kazman et al. (1995).

7 Cf. Becker, Rosemann, Schuette (1995), p. 466.

8 Cf. Hars (1994), pp. 7-11.

9 Cf. Hars (1994), p. 11. cf. Nissen et al. (1996), p. 38.

10 Cf. Ferstl, Sinz (1994), p. 86; Steele, Zaslavsky (1994), p. 317: "Meta models [...] might be expressed using one or more modelling techniques, that in combination are able to adequately model all relevant aspects of any given modelling technique."

11 Cf. Saeki (1996).

2.2 Meta models for workflow management systems

In the past a number of different meta models for workflow management systems have been introduced. In most cases these models are reference meta models. Examples for such reference meta models can be found in DERUNGS ET AL. (1995), GALLER (1995), JABLONSKI, BUSSLER (1996) and JOOSTEN (1996).

The aim of this paper is not the introduction of another reference meta model but the evaluation of product individual meta models as well as the discussion of the results delivered by the comparison of these (detailed) meta models. The integration of the presented product individual meta models into an - inductively generated - reference meta model is another step to be taken. This reference meta model is supposed e.g. to serve as a foundation for the design of a process information system, used for process monitoring and controlling, as well as a guideline for information system designers deriving meta models from given systems.

2.3 Description of the notation

The meta data models in this paper are designed using extended Entity-Relationship Models.[12] An extension of this method are variable-based integrity constraints.[13] These allow instance specific definitions of semantic relationships (e.g. exclusive-OR) between cardinalities. This is achieved by replacing the cardinalities with a variable and relating the cardinalities in separate integrity constraints. Furthermore, generalizations are established for reasons of clarity, e.g. groupings of all relevant information objects that are relevant for the staff resolution. For every generalization a description of the disjointness constraint (d (disjoint) or n (not disjoint)) and the completeness constraint (p (partial) or t (total)) is given.[14]

3 Comparison of meta models

3.1 Conflict identification and resolution

Prior to the evaluation of meta models it has be made sure that all possible conflicts between the models have been resolved. Three major kinds of conflicts can be identified:[15]

Naming conflicts occur, if the naming conventions of the models to be compared show synonyms or homonyms. They are the most common conflicts during the evaluation of meta data models. A synonym can be found if two or more linguistic expressions share the same meaning. A homonym results of the ambiguity of a linguistic element. In its context the meaning of a homonym is unique, but not because of its notion alone.

The determination of *synonyms* asks for an analysis of different terms with identical meaning and associations. Hints about potential synonyms can be found by tracing similar structures embedding information objects with different names (concept likeness)[16]. *Homonyms* contradict the clarity of a model, because the notion of the term can be determined depending on the user and the context, but not in general. A potential indicator for the identification of homonyms in meta models are information objects with the same name that are embedded in different structures (concept unlikeness).

A *type conflict* can be located, if the same fact is represented semantically correct in two models through different methodical concepts. *Structural conflicts* arise if the meta models to be integrated depict the same facts using different semantics. This is a violation of the semantic correctness. Structural conflicts often arise if different people are involved in the modeling process. Type and structural conflicts are not relevant at this point, because the meta data models to be compared and to be evaluated were created by the same group of people.

---

12 Cf. Chen (1976).

13 Cf. Rosemann (1996).

14 Cf. Elmasri, Navathe (1995), p. 618. They use the term overlapping (o) instead of not disjoint.

15 Cf. Hars (1994), pp. 194-206.

16 Cf. Batini, Lenzerini (1984); Batini, Ceri, Navathe (1992), p. 124.

3.2 General aspects for the evaluation of meta models

When two conflict-free meta data models are compared , the following information objects can provide useful information:

Entity types

The comparison of the number and kind of entity types provides the most essential information for the comparison of meta data models. The adaptability of a workflow management system increases with the number of entity types the meta data model consists of within a given degree of abstraction. The validity of a qualitative and quantitative rating of entity types is determined by the degree of detail of the models in the first place.

Relationship types

Another source of information about the flexibility of a workflow management system are the relationship types. The flexibility of a workflow management system increases with the number of relationship types, if the number and kind of entity types stay the same. An example for different relationship types can be found in the organizational model of a workflow management system (chapter 3.3), where the relations between actors, organizational units and roles differ between the systems.

Cardinalities

The cardinalities provide useful information about the design options a specific system provides. A workflow management system, that allows a (0,1)-(0,n)-hierarchy of organizational units, simultaneously prohibits the design of multi-dimensional organizations (e.g. matrix-structures). The introduction of variable-based cardinalities enhances the semantic capability of a model, but complicates the comparability of the model.

Attributes

Beyond the elements an ERM usually provides, the comparison of attributes provides further information about the system characteristics a workflow developer can use. Attributes have to be distinguished into attributes used by the system and additional ('free') attributes. User defined attributes allow a higher flexibility than predefined attributes.

3.3 Evaluation of the organizational meta models

Before the meta data models are compared, a brief description of the central entity types is given. Potential naming conflicts have already been resolved during the modeling of the two Entity-Relationship-Diagrams depicted in fig. 1 and 2. For example, the entity type *role* of WorkParty was renamed *position type*, while the entity type *compentence* was renamed *role* in order to avoid a homonym conflict with the corresponding terms of FlowMark.[17]

An *organizational position* in WorkParty is the domain of a person within an organizational unit and is assigned to exactly one organizational unit. An organizational position can be occupied by one person, but it may be left empty, too. In WorkParty an organizational position is an abstraction of a specific person. Zero, one or several persons can act as a substitute for a specific organizational position. An organizational position can belong to a certain position type. This entity type cannot be found in the meta model of FlowMark.

---

17 One can identify a homonym conflict between the entity type *role* of WorkParty and FlowMark as well as a synonym conflict in the renamed entity type *competence* and the original entity type *role* of WorkParty. A correct resolution of these conflicts asks for a new context free term for the entity types concerned. However, the term *role* has a generally accepted meaning in this context and is used by the WfMC as well. Therefore, in this particular case the conflict resolution is passed up in order to use generally accepted terminology.

An *organizational unit* in WorkParty is a part of the company organizational structure, e.g. a department. It is created as a composition of organizational positions or subordinate units. Organizational units can be temporary or permanent, the former are called project units, the latter are called line units and form the core of the company organizational structure. The hierarchical structure of organizational units is restricted to a tree structure. An organizational unit may be associated with several roles and positions and can be the owner of several resources. Similar to WorkParty, an *organization* in FlowMark represents an administrative unit of the company organizational structure. However, a person in FlowMark can only be member of one organization. An organization has one specific manager and may have an unlimited number of members with different roles, but at least one (the manager).

Resources (only in WorkParty) are items that ultimate units of responsibility may dispose in order to fulfill their tasks. Each resource may be assigned to one or more roles. This leads to the definition of access rights for certain persons. Each Resource has one specific owner, which may be a person, an organizational unit, or an organizational position. Resources can be organized in an is-part-of-relationship, thus being superior or subordinate to other resources.

Levels (only in FlowMark) represent a hierarchy inside the organization that need not be identical with the company organizational structure (e.g. years of membership or tariff).

When compared directly, the WorkParty meta model shows a larger number of entity types than the FlowMark meta model, which points to a higher flexibility of modeling different organizational structures. FlowMark misses especially the entity types *organizational position* and *position type*. In contrast to this, FlowMark offers two relationship types between the entity types *person* and *organization* or *person* and *role*, while WorkParty offers only one relationship type between these entity types.

FlowMark shows more restricted cardinalities than WorkParty. Especially the rigid (1,1):(1,n)-relationship between the entity types *person* and *organization* is expected to be a handicap when modeling organizations. The possibility of assigning several *positions* to one *person* in WorkParty enables the modeling of multi-dimensional organizations and shows greater flexibility than the solution provided by FlowMark. Both products show a restricted (0,1):(0,n)-hierarchy over the entity type *organization*. In addition, the substitute-relationship for the entity type person in FlowMark is restricted to only one substitute per person ((0,1):(0,n)-relation), while WorkParty allows an unlimited number of substitutes. In WorkParty a person has to be assigned to a position and therefore to a specific organizational unit. This prohibits the modeling of external workers, such as contractors.

The interface between the organizational meta model and the process meta model is provided by an *address expression*. This entity type enables a dynamic staff resolution at runtime, e.g. according to the criteria given in the address expression and the current system users, all persons are determined, that match the given criteria. In general, direct and indirect address expressions can be distinguished. While the former explicitly reference the authorized actors in person, the latter contain criteria like organizational units or positions which are used to determine a subset of authorized persons at runtime. The use of indirect address expressions makes processes more resistant against changes in the organizational model of a company than it would be possible with a direct assignment of persons to activities.

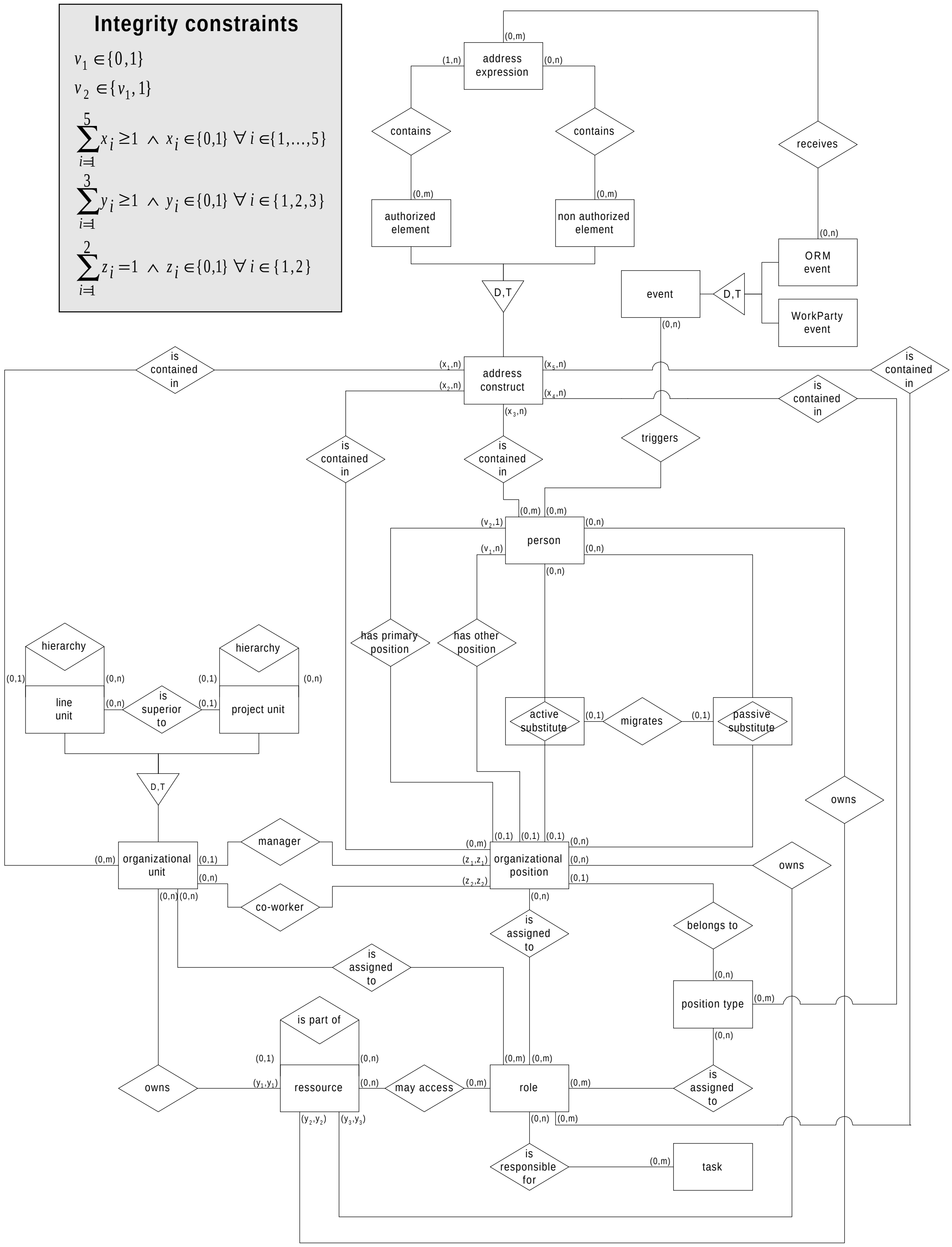

Fig. 1: Organizational meta model of WorkParty

Integrity constraints

$$\sum_{i=1}^{2} x_i \geq 1 \wedge x_i \in \{0,1\} \forall i \in \{1,2\}$$

$$\sum_{i=1}^{3} y_i \geq 1 \wedge y_i \in \{0,1\} \forall i \in \{1,2,3\}$$

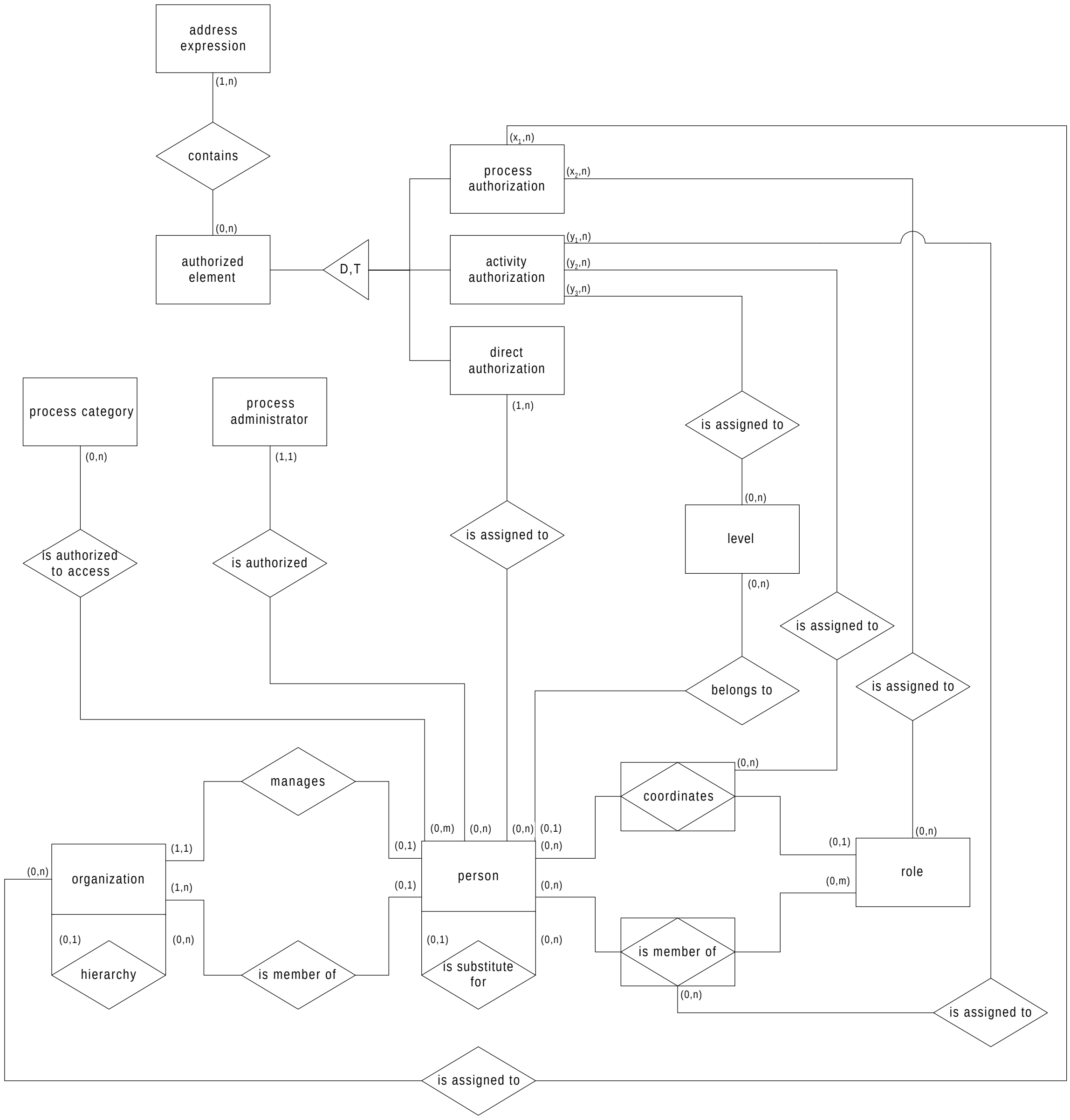

Fig. 2: Organizational meta model of FlowMark

3.5 Evaluation of the process meta models

Both FlowMark and WorkParty provide the user with a graphical modeling method for designing workflow models. In FlowMark workflow models are drawn as graphs with weighted transitions through the explicit design of knots and connectors. WorkParty provides the user with pre-defined process components. These components provide elementary control flow elements, such as iterations, forks and joins.

3.5.1 The process meta model of WorkParty

In WorkParty a *process* is a directed graph that determines the sequence of activities or programs respectively. It consists of a starting activity and one or more ending activities, knots and connectors. The starting activity contains global information about the process that are relevant for all activities. Knots contain activities or programs, connectors determine their sequence. Several ending activities can be used to model alternative endings of a process, the starting activity of a process has to be unique.

An *activity* is an elementary step in a process. Activities can be manual or they can reference an application program. Furthermore, programs may be associated with activities that are only executed in case of an exception. WorkParty supports ad-hoc activities, i.e. at runtime the user may insert an unlimited number of new activities at a specified point, before the modeled process is resumed.

A *file* is a database used by WorkParty for storing process and activity templates. While a theme file contains all process and activity templates that are related to a specific field of business, a template file is a subset of a theme file and contains process and activity templates for a certain kind of business process. Process files are instances of template files and contain attachments and parameters relevant for the specific processes.

3.5.2 The process meta model of FlowMark

A *process* in FlowMark is similar to a process in WorkParty. In addition to the sequence of the activities the process designer in FlowMark has to specify the flow of data between the activities through the use of data connectors. Each process can be assigned to one process category in order to restrict the number of people authorized to execute the process. Furthermore, a process may contain an address expression that is valid in addition to activity individual address expressions. For every process a process administrator can be defined, who is contacted in case of an exception.

An *activity* is a single step in a process and corresponds to the activity in WorkParty. Activities contain a start condition and an end condition as well as an address expression for the individual staff resolution. FlowMark supports three kinds of activities:

- *Process activities* are used to invoke sub processes within a process. It is possible to recursively invoke the same process.
- *Program activities* are (semi-)automated activites, that invoke an application program during their execution.
- *Loop elements* are used for modeling iterations. If a single activity shall be executed several times the end condition of the activity can be used as a loop condition. If the iteration shall contain several activities a *block* has to be used. Blocks are equivalent to sub-processes, but their contents are modeled individually. *Bundles* are used if a single activity has to be instantiated several times.

Containers are used for storing data that is transmitted from one activity to another. Each activity, process and block has one input container and one output container.

A *connector* is a directed edge that determines the control flow and the data flow of a process. Control connectors can have a transition condition that must be fulfilled, otherwise the target activity will not be executed and the path of the connector will be eliminated during a dead path elimination. Default connectors have no transition condition and are followed if none of the other control connectors evaluates to true. Data connectors are used for connecting the input- and output containers of activities. The flow of data in FlowMark need not to be identical with the control flow.

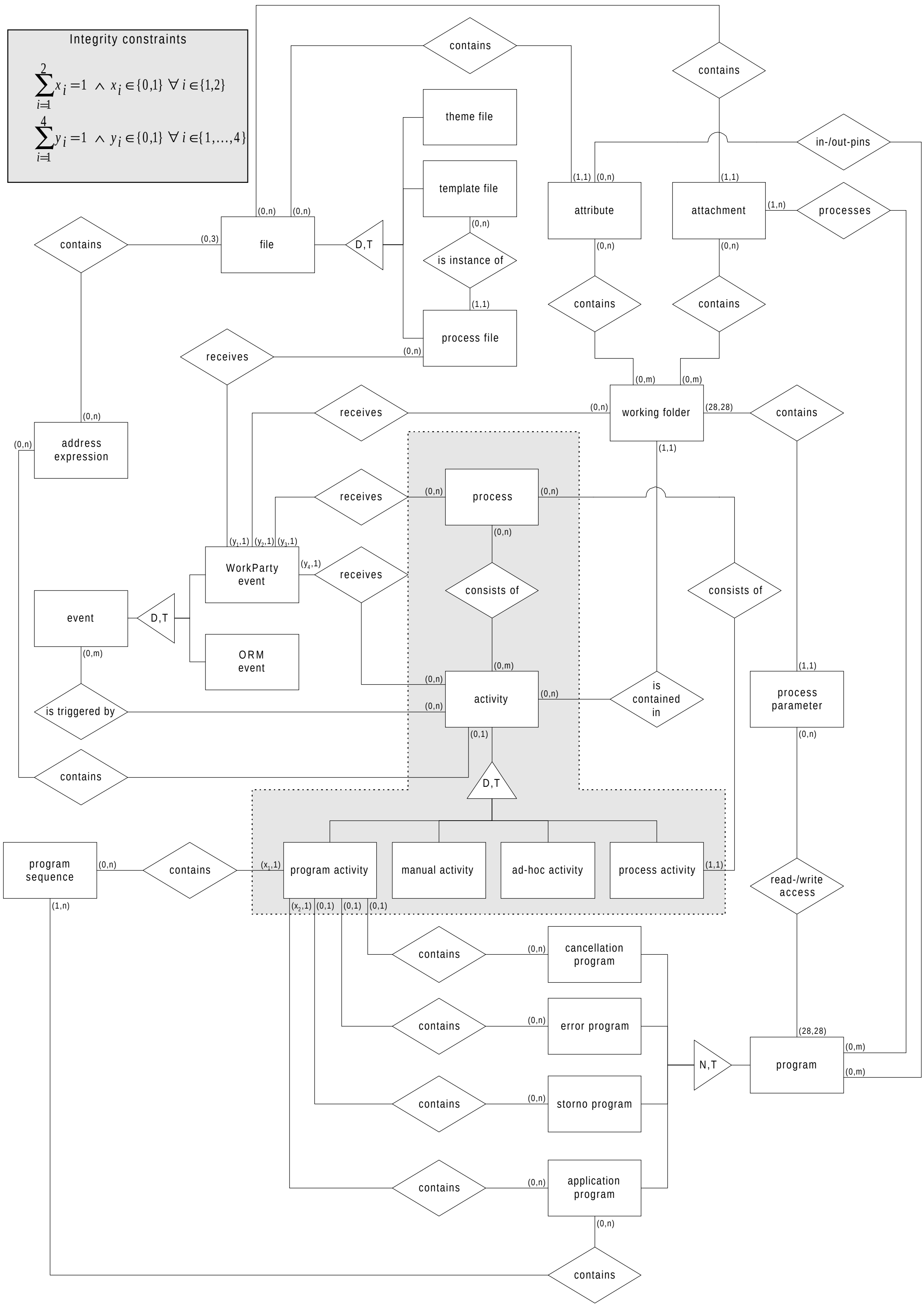

Fig. 3: Process meta model of WorkParty

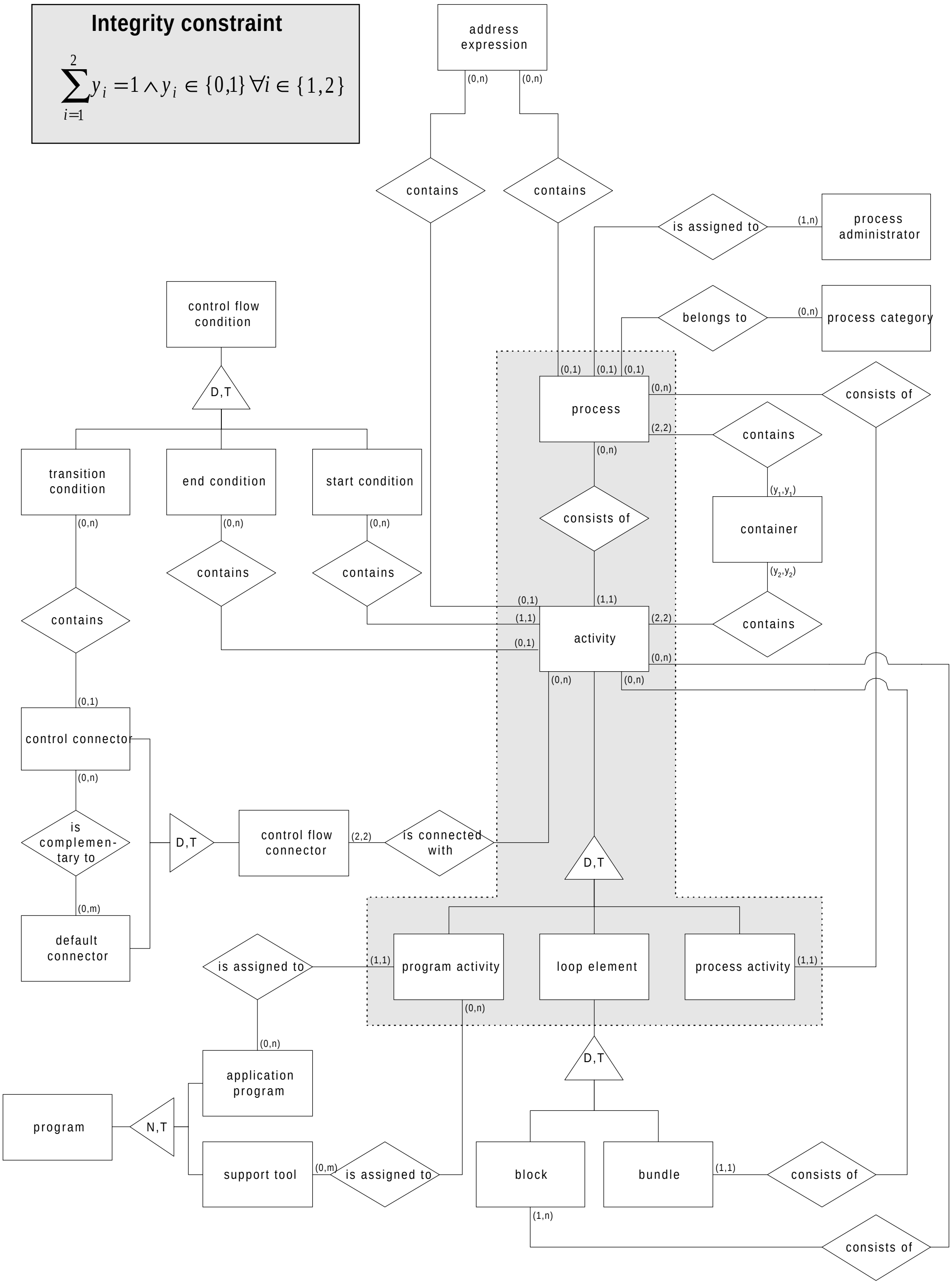

Fig. 4: Process meta model of FlowMark

3.5.3 Evaluation of the process meta models

Similar to the conflict resolution during the comparison of the organizational meta models several conflicts had to be resolved before the process meta models could be compared. For example the entity type *process* of FlowMark is called *sequence* in WorkParty. The resolution of conflicts was done by the selection of WfMC-compliant terms. The gray reverse *T* shapes in figure 3 and 4 indicate characteristic design elements that can be found in the process meta models of different workflow systems.

In FlowMark control and data flow are modeled separately and are therefore visible as distinct elements. In WorkParty the control flow is modeled by customizing predefined components while the data flow is not visible in the process model. Instead, the data flow is modeled for each activity individually. The control flow elements available in both systems are not completely represented in the meta models. For example, the bundle activity of FlowMark can be implemented in WorkParty as well, using certain iteration components in combination with optional process branches. Therefore, the semantic power of the meta models cannot be compared without taking the control flow elements into account.

Contrary to the organizational meta model a direct comparison of the number of entity types does not seem reasonable for the process meta models. Because of the different workflow philosophies the mere number of entity types does not lead to conclusions about advantages or disadvantages of a specific system. The selection of the appropriate level of detail is difficult, too, because on the one hand the meta models should not be subject to unlimited growth and on the other hand sufficient aspects for comparison should be available.

The great variety of workflow modeling methodologies shown by both systems makes the comparison of the process meta model a lot more difficult than the comparison of the organizational meta models. Therefore, an enhancement of the evaluation with a catalogue of basic control flow elements seems to be useful. The results of such an enhancement are a starting point for the evaluation of the semantic modeling power of different workflow management systems.

4 Summary and future directions

A comparison of meta models can provide useful information for the evaluation of systems, but it depends on the context of the analysis. *Organizational meta models* of workflow management systems are limited to a well documented domain with little product-individual differences. In this context a meta model evaluation provides the analyst with important information about the modeling capabilities of the methods analyzed. *Process meta models*, however, are only one instrument during the evaluation of workflow management systems. They are useful for the analyst to gain an overview about the modeling methodologies and to get acquainted with the individual systems, but they are not sufficient for a final decision for or against a specific system.

Some extensions for existing modeling methodologies have been introduced, such as variable-based cardinalities, which enhance the suitability of these models. Future work should emphasize this aspect. Further fields of research are - besides the inclusion of other workflow management systems - the design of an inductively generated reference meta model as well as the development of a catalogue of basic control flow elements.[18]

References

Bartholomew, D.: A Better Way To Work. Workflow systems, when combined with business reengineering, speed tasks and processes. In: Information Week 11.9.1995, pp. 32-40.

Batini, C.; Ceri, S.; Navathe, S. B.: Conceptual Database Design. An Entity-Relationship-Approach. Redwood City et al. 1992.

Batini, C.; Lenzerini, M.: A Methodology for Data Schema Integration in the Entity Relationship Model. IEEE Transactions on Software Engineering, 10 (1984) 6, S. 650-664.

Becker, J., Rosemann, M., Schuette, R.: Grundsaetze ordnungsmaessiger Modellierung. Wirtschafts-informatik, 37 (1995) 5, pp. 435-445. (in German)

Chen, P. P.-S.: The Entity-Relationship Model: Toward a Unified View of Data. ACM Transactions on Database Systems, 1 (1976) 1, pp. 9-36.

Derungs, M., Vogler, P., Österle, H.: Metamodell Workflow. Report Nr. HSG/CC PSI/3. Version 1.0.http://www-iwi.unisg.ch/iwi2/cc/psi/abpfi3.html. St. Gallen 1995. (in German)

Dunn, R. H.: Software-Quality. Concepts and Plans. Englewood Cliffs 1991.

18 A catalogue of control flow elements and the analyses of two additional workflow management systems can be found in zur Muehlen (1996).

Elmasri, R.; Navathe, S. B.: Fundamentals of Database Systems. 2nd Ed. Benjamin/Cummings Publishing, 1995.

Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik. Band 1. 2nd Ed., Vahlen, 1994. (in German)

Galler, J.: Metamodelle des Workflow-Managements. Veroeffentlichung des Instituts fuer Wirtschafts-informatik Nr. 121. Ed.: A.-W. Scheer. Saarbruecken 1995. (http://www.iwi.uni-sb.de/ public/iwi-hefte/iwihef_1.html) (in German)

Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow-Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases, 3 (1995) 2, pp. 119-153.

Hars, A.: Referenzdatenmodelle. Grundlagen effizienter Datenmodellierung. Gabler, 1994. (in German)

IBM FlowMark: Modeling Workflow. Version 2 Release 3. Document Number: SH19-8241-02. Ed.: IBM Deutschland Entwicklung GmbH. Wien 1995.

Jablonski, St.; Bussler, C.: Workflow-Management: Modeling Concepts, Architecture and Implementation. International Thomson Computer Press, 1996.

Joosten, S.: Conceptual Theory for Workflow Management Support Systems. Technical Report. Centre for Telematics and Information Technology. University of Twente. Enschede 1995.

Joosten, S.: WorkPAD: a Conceptual Framework for Process Analysis and Design. In: ACM Transactions on Office Information Systems (under submission). Atlanta 1996.

Kazman, R.; Bass, L.; Abowd, G.; Webb, M.: SAAM: A Method for Analyzing the Properties of Software Archietectures. In: Proceedings of the International Conference on Software Engineering - ICSE'16. Ed.: B. Fadini. Sorrento 1994, pp. 81-90.

Leymann, F.; Altenhuber, W.: Managing business processes as information resources. IBM Systems Journal, 33 (1994) 2, S. 326-348.

zur Muehlen, M.: Der Loesungsbeitrag von Metamodellen und Kontrollflussprimitiven beim Vergleich von Workflowmanagementsytemen. Diploma Thesis. Department of Information Systems. University of Muenster 1996. http://www-wi.uni-muenster.de/is/lehre/diplom/ismizu. (in German)

Nissen, H. W., Jeusfeld, M. A., Jarke, M., Zemanek, G. V., Huber, H.: Managing Multiple Requirements Perspectives with Metamodels. IEEE Software, 13 (1996) 3, pp. 37-48.

Rosemann, M.: Komplexitaetsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungs-empfehlungen fuer die Informationsmodellierung. Wiesbaden 1996. (in German)

Saeki, M.: Object-Oriented Meta Modelling. In: Proceedings of the 14th int. Conference on Object Oriented and Entity Relationship Modelling. Ed.: M. P. Papazoglou. Berlin et al. 1995, pp. 250-259.

Siemens Nixdorf: Organisations- und Ressourcenmanagement Version 2.0, Referenzhandbuch, Ausgabe Juni 1995, Muenchen 1995. (in German)

Siemens Nixdorf: WorkParty Benutzerhandbuch Version 2.0, Muenchen 1994. (in German)

Steele, P. M., Zaslavsky, A. B.: The Role of Meta Models in Federating System Modelling Techniques. In: Proceedings of the 12th International Conference on the Entity-Relationship Approach - ER `93. Eds.: R. A. Elmasri, V. Kouramajian, B. Thalheim. Berlin et al. 1994, pp. 315-326.

WfMC: Workflow Management Coalition Terminology & Glossary. Document Nr, WFMC-TC-1011. 2nd ed. Bruessel 1996.