Evaluation of Workflow Management Systems Using Meta Models

Michael zur Mühlen
Department of Information Systems, University of Muenster
Steinfurter Str. 107, D-48149 Muenster, Germany
Tel.: +49 (251) 8338080, Fax: +49 (251) 8338109
ismizu@wi.uni-muenster.de

Abstract

The analysis of workflow meta models aims at a problem that frequently occurs during the selection of a workflow management system – evaluating the capabilities of a modeling method. In this paper a meta model approach for the evaluation of different workflow management systems is introduced. After a comparison of the meta models of current workflow management systems a procedure model for the evaluation process is specified. An organizational reference model is introduced, which helps users in specifying their requirements for a workflow management system. A comparison of scoring model- and meta model-based evaluation processes concludes this paper.

1. Introduction

The increasing number of workflow vendors (86 at the 1998 CeBIT Fair) and the variety of their products creates a difficult situation for potential workflow users. On the one hand, the likelihood of an appropriate solution for the current business problem increases, on the other hand, the evaluation of the different systems is becoming a time-consuming and costly process. While a lot of work has been put into the development of standardized scoring models for software evaluation, only a few researchers focus on a specific methodology for the selection of workflow products. In this paper we present a meta model-based approach for the evaluation of workflow management systems. After a description of the traditional process of software evaluation, the meta model based evaluation process is introduced in section 4. Section 5 demonstrates the application of the proposed methodology through the comparison of two workflow management systems - *IBM FlowMark 2.3* (Version 3.0 is entitled MQSeries Workflow) and *SNI (Siemens-Nixdorf) WorkParty 2.0*. In the sixth section, an organizational reference model is introduced that can serve as a benchmark for existing workflow management systems. In the last part of the paper, a comparison of scoring model- and meta model-based evaluation processes outlines which aspects of workflow systems should be evaluated using the different approaches.

2. Criteria catalogues for software evaluation

The selection of a software system from a number of alternatives demands a valuation of each single product by the user. Qualitative and economical aspects are the two main groups of criteria for the selection of a software product. DUNN identifies four quality aspects of software: Reliability, Usability, Maintainability and Adaptability [1]. A different classification can be found in [2], where three groups of quality criteria are identified:

- Quality aspects that determine the *output* of the system analyzed. These are – among others – correctness, security, reliability and availability.
- Quality aspects that determine the *activities of the developer* and/or the maintenance personnel. This category includes maintainability, portability, scalability and adaptability.
- Quality aspects that are *relevant for the user* during the use of the system. These are e.g. ease of use/ usability, predictability and learnability.

These quality criteria mainly affect the technical aspects of the software evaluated. During the evaluation of a software system economic issues are considered through the use of economic criteria. These are e.g. the purchase cost of the system, installation effort, necessary hardware, training cost for users and maintenance cost. Further criteria for the selection of a specific software system can also affect the decision, e.g. the market share of the product, the reputation of the manufacturer and existing business relations.

Not all of the criteria mentioned are of same importance for all users. Therefore during the selection of a software system "criteria catalogues" are used, which consist of a selection of criteria relevant to the specific purpose. In the literature, a number of pre-defined criteria catalogues exist, which can be customized by the elimination of irrelevant criteria and the refinement of relevant criteria (cf. e.g. [3]). The scores resulting from a use of the criteria catalogue can be valued differently, depending

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

on the kind of criteria analyzed. A distinction can be made between methods that analyze a system based on a single criterion, e.g. cost (one-dimensional scoring methods) and scoring methods, that analyze an alternative based on a mixture of several criteria, e.g. by calculation of the cost/risk relation (multi-dimensional scoring methods).

However, the use of a criteria catalogue is not recommended in all cases. Some criteria cannot be measured using a metric scale, e.g. the security of a program. In order to determine the value of these criteria, a relative scaling based on benchmarks is used. In this case the (by subjective selection) best product available is used as a benchmark for the evaluation of the alternatives [4]. In these cases the choice of the scale and the measurement of deviations in an objective way can be a problem. Sometimes it is impossible to determine the criteria at all, e.g. the correctness of a large-scale program cannot be determined for every situation possible with reasonable effort. As a substitute, the criterion robustness can be evaluated instead, e.g. it is checked, whether the program shows predictable reactions during all possible forms of external interaction. An option to reduce the complexity of evaluation criteria is the use of a scenario technique, e.g. SAAM [5]. In the field of workflow management scenario-based evaluations have been performed with little success (cf. e.g. [6]).

3. Evaluation of workflow management systems

During the analysis of a workflow system, the ability to integrate the company's existing and new business processes is of great relevance. Therefore, the quality of a workflow management system is not only determined by the quality of the system itself, but also by the quality of the workflow models that can be modeled using the system. This aspect is even more important, if the projected lifetime for a workflow system is taken into account. Contrary to desktop application programs with a fairly short lifecycle, workflow management systems are expected to have a lifecycle that may easily last for more than a decade (the average age of existing data base systems in enterprises and the foreseen increase of software age lead to this conclusion, cf. [5]).

If the same criteria catalogue are used for the evaluation of workflow systems as for e.g. ERP-Software, some problems arise, e.g. the choice of a favored evaluation dimension. Evaluations based on criteria catalogues show a vertical and a horizontal dimension. STRAHRINGER distinguishes horizontally-dominant and vertically-dominant evaluation schemes [7]. During a horizontally-dominant evaluation the products are analyzed against a number of elementary criteria, which show a high degree of structure. The integration of additional products in such a scheme is possible with little effort, since the criteria show answers such as „yes/no“ or „available/not available“, which can be determined in an easy fashion. Contrary to this, a vertically-dominant comparison puts a thorough analysis of the single products into focus. In this case, complex criteria are used that are described using natural language. Compared to horizontally-dominant catalogues, this kind of comparison involves fewer criteria, which are less structured due to their complexity. The addition of new products is more complicated, because the analysis of the single aspects leads to a more complex analysis of the systems.

An evaluation of modeling methods for workflow management systems based on a horizontally-dominant scheme carries the risk that the complexity of the methods analyzed is not represented in the evaluation scheme sufficiently. The selection of a vertically-dominant scheme leads to a less formalized evaluation, which makes a comparison of the results difficult.

Meta model based evaluations are a compromise between horizontally- and vertically-dominant schemes. Through the use of meta models, complex evaluation criteria can be expressed using a higher degree of formalization than it would be possible using natural language[7], without having a negative effect on the completeness of the analysis. SCHWAB points out the importance of meta-model-based evaluations of workflow systems during his classification of systems using coordination principles [8]. For a comparison of workflow systems based on meta models see [9] and [10].

4. System evaluation based on meta models

The analysis of workflow meta models aims at a problem that is inherent with the nature of the selection of a workflow management system - evaluating the capabilities of a modeling method. Since workflow management systems provide users with facilities for modeling organizations and processes, the evaluation of these modeling methods is a crucial part of the selection process. Most workflow management systems are divided into a modeling component (buildtime) and a runtime client. The buildtime component is used for the modeling of organizational entities as well as the definition of the workflow models. Since the capabilities of this part have a great influence on the quality of the later workflow models, its quality has to be analyzed closely, before a system can be chosen.

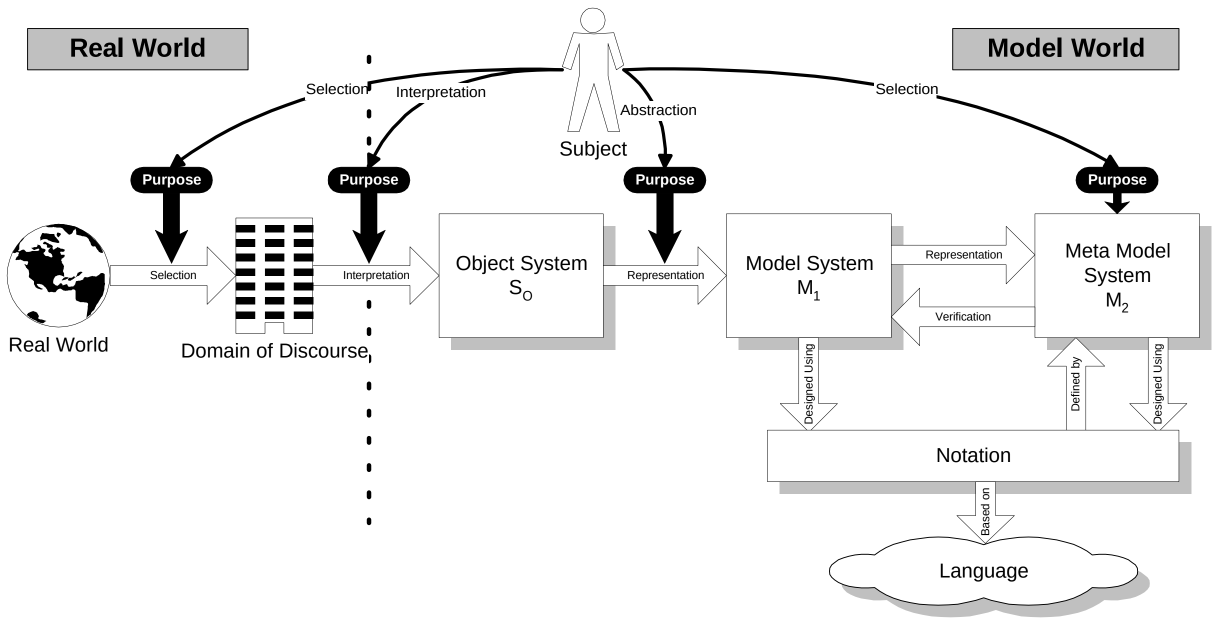

Figure 1. Relationship between model and meta model systems

4.1. Characteristics of meta models

A model is an immaterial representation of an object system. It is created for the purpose of a subject, relates two systems, and therefore consists of three components:
The *object system* represents the subjective interpretation of a selected part of the real world (domain of discourse) including the relevant part of the environment.
The *model system* represents the subjective representation of the object system. A syntax (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 (cf. figure 1, [11]). 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 semantics. This framework also defines rules for the use and specialization of model elements and relationships. (cf. [12] 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.").

In the context of information modeling we distinguish meta models describing a notation and meta models describing a procedure [13]. *Meta data models* characterize notations that can be used for information modeling purposes, in our case the modeling methodology of a workflow management system. *Meta process models* describe the modeling process using a specific method, in this case the procedure of creating a workflow model using a specific workflow management system. Every meta model is based upon another meta model, which can be of the same kind, e.g. the notation of the entity relationship diagram may be explained using another entity relationship 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*.

4.2. 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 (cf. e.g. [14]).

The main objective 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 a reference meta model is outlined in section 6.

4.3. Description of the notation

The meta data models in this paper are designed using extended Entity-Relationship Models. Variable-based integrity constraints are an extension of this method [15]. 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. Although variable based integrity constraints raise the complexity of the meta data models, they allow the reduction of the number of different elements and therefore raise the clarity of the meta data models. 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 O (overlapping)) and the completeness constraint (P (partial) or T (total)) is given.

5. Comparison of meta models

5.1. Identification and resolution of conflicts

Prior to the evaluation of meta models it must be determined that all possible conflicts between the models have been resolved. Three major kinds of conflicts can be identified:

Naming conflicts occur, if the naming conventions of the models to be compared show synonyms or homonyms. These are the most common conflicts during the evaluation of meta data models. A homonym exists, if *different* elements of two or more workflow management systems have the same name. A synonym can be found, if *the same* element of two or more workflow management systems are identified using different expressions.

The determination of *synonyms* requires 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) (cf. [16]). *Homonyms* contradict the clarity of a model, because the meaning 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 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.

5.2. Information generated by a meta model evaluation

When two 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 embedded in the meta data model 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 (section 4.3.), where the relations between actors, organizational units and roles differ between the systems.

Cardinalities: Cardinalities provide useful information about the design options of a specific system. A workflow management system that allows the modeling of 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 makes comparisons of the models more complex.

Attributes: Beyond the elements that 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 between attributes used by the system and additional ('free') attributes. User definable attributes increase the flexibility of a system.

5.3. Evaluation of the organizational meta models

Before the meta data models of IBM FlowMark and SNI WorkParty 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. 2 and 3. For example, the entity type *role* of WorkParty was renamed *position type*, while the entity type *competence* was renamed *role* in order to avoid a homonym conflict with the corresponding terms of FlowMark.

A homonym conflict can be identified 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 Workflow Management Coalition (WfMC) as well. Therefore, in this particular case the conflict resolution is passed up in order to use generally accepted terminology.

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.

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 a member of one organization. An organization has one specific manager and may have an unlimited number of members with different roles, but must have at least one (the manager).

Resources (only in WorkParty) are items that ultimate units of responsibility may consume 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).

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 in modeling different organizational structures. FlowMark lacks 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.

5.4. 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. For a the Entity-Relationship-Diagrams of the process meta models cf. [9].

Integrity constraints

$$v_1 \in \{0,1\}$$

$$v_2 \in \{v_1, 1\}$$

$$\sum_{i=1}^{5} x_i \geq 1 \wedge x_i \in \{0,1\} \; \forall \; i \in \{1,...,5\}$$

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

$$\sum_{i=1}^{2} z_i = 1 \wedge z_i \in \{0,1\} \; \forall \; i \in \{1,2\}$$

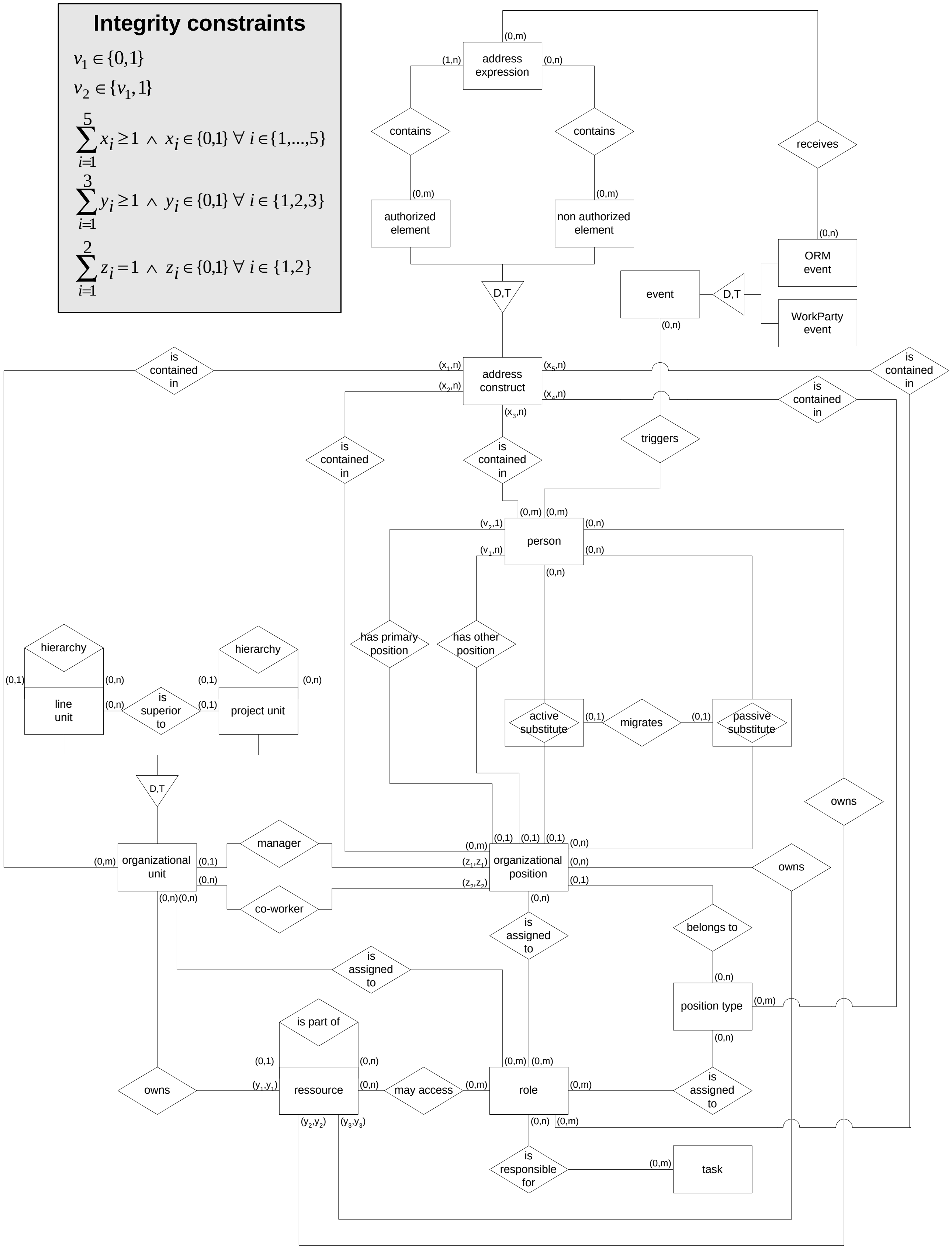

Figure 2. Organizational meta model of WorkParty

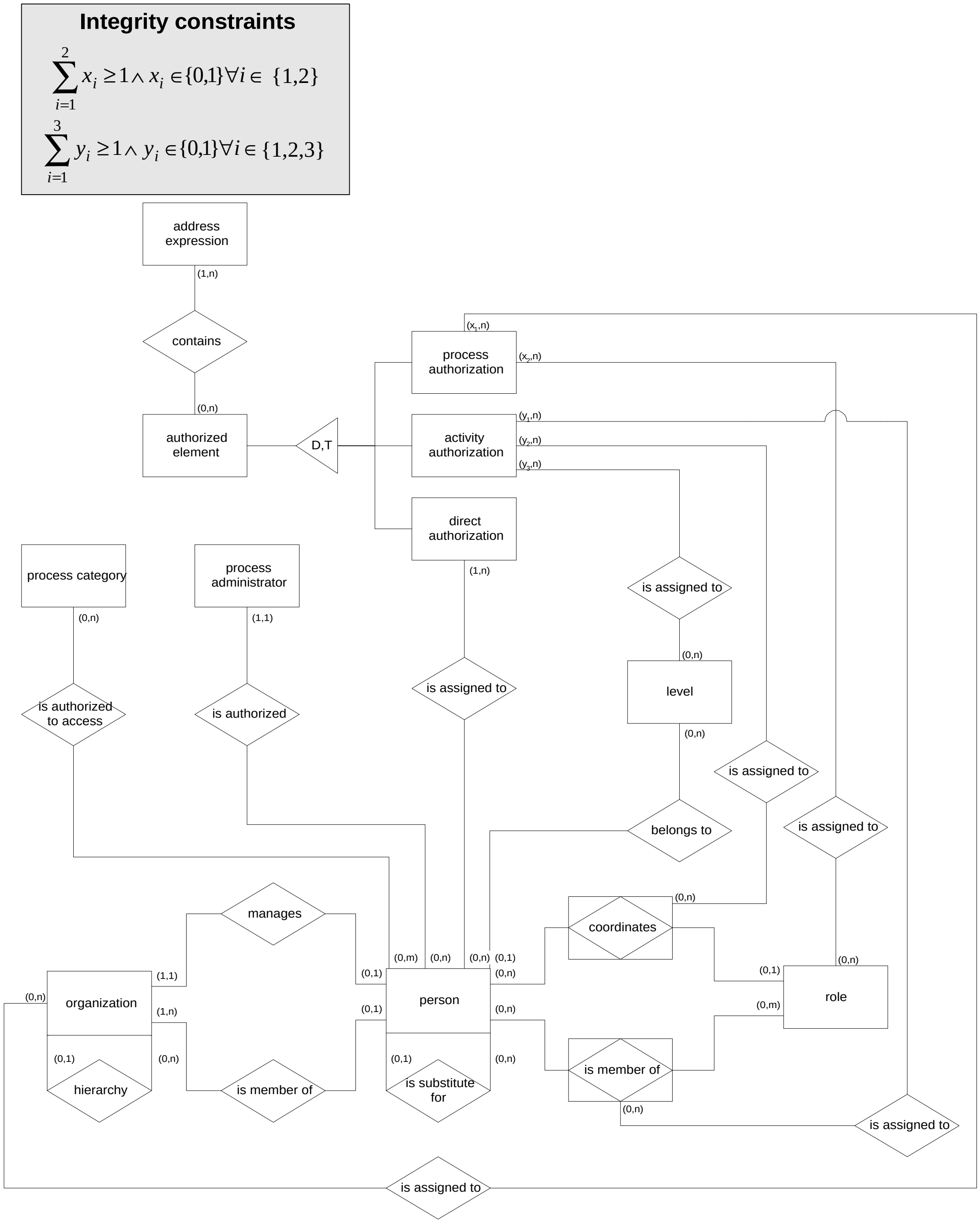

Figure 3. Organizational meta model of FlowMark

5.4.1. 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 is 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.

5.4.2. 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 activities, 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 contains 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 structured 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 no transition condition 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.

5.4.3. Comparison 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.

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 modeling methodologies the mere number of entity types does not allow 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. For an analysis of the control flow elements of the systems analyzed cf. [17].

Another important aspect is the flexibility of the workflow management system, which can be divided into the flexibility of the modeling methodology used for the creation of the workflow models (buildtime flexibility) and the runtime flexibility that can be achieved e.g. by a change of the workflow model during the execution of a workflow instance. Criteria for this kind of flexibility include e.g. the possibility to add resp. skip activities at run-time, the option to select whether the change affects the running instance or only future instances and the possibility to delegate work. Since an evaluation based on process meta models only takes into account the flexibility of the buildtime models, the runtime flexibility should be evaluated using a suitable scoring model. The demand for runtime flexibility strongly depends on the characteristics of the business process that is subject to workflow automation. If the process is not likely to change (e.g. due to some legal restrictions) runtime flexibility is less important but if not every activity of the process can be formalized the runtime flexibility of the workflow management system has to be represented in the scoring model.

6. Reference meta models as a benchmark for meta data models

6.1 An organizational reference model for workflow systems

The evaluation of the meta data models described above as well as the analysis of two additional workflow management systems led to the conclusion, that an independent benchmark had to be established in order to enable a more objective analysis of workflow systems. (The details of the analysis can be found in [17]). The additional systems analyzed were CSE/WorkFlow 4.1 and o.tel.o LEU. While the process meta models of these systems showed certain similarities as described above, the organizational meta models varied widely and they also served as input for the development the reference meta model outlined here.

The evaluated workflow systems showed substantial weaknesses regarding their organizational modeling component, whereas the process modeling capabilities were much more sophisticated. Therefore an organizational reference model has been developed at the University of Muenster, which serves as a benchmark for the evaluation of meta data models (figure 4). This reference meta model includes several aspects which enhance the ideas found (or missed) in the systems analyzed. The main objective during the development of the reference meta model was to obtain a maximum of flexibility and adaptability in order to use the model for a broad variety of organizations. Consequently, the meta data models of existing workflow systems served as a starting point for the reference model, but it finally turned out that a flexible meta model could be designed with a relatively small number of entity and relationship types.

6.2. Additional entity types

Most workflow systems provide the modeler with a role concept for the modeling of organizational entities. Roles serve as a container for the qualifications an actor has to have in order to fulfill a given task. However, in real life organizations there is a difference between qualifications and competencies. While the former are attributes of a specific actor and cannot be removed (i.e. “Spanish Language Skills”), the latter depend on the position the actor occupies in a given organization (i.e. “may sign orders over $ 50.000”) and can be assigned and removed. Consistently, the entity type *role* is subdivided into the two entity types *qualification* and *competence*.

The modeling of committees, expert groups and other temporary organizational units is enabled by two elements. First, the strict organizational hierarchies [(0,1)–(0,n)] found in WorkParty and FlowMark are replaced by *structures* [(0,n)–(0,m)], which allow the modeling of multidimensional relationships between roles, positions and organizational units. Secondly, the relationship between actors and organizational units is separated by the introduction of *positions*, which form the static organizational structure of a company. A person can occupy several organizational positions and can therefore be a member of several organizational units, those being *temporary* (e.g. projects) or *permanent* (e.g. departments).

Since an organizational position serves as a placeholder for a specific person, the competencies related to a position should be assigned to the position itself instead of to the person occupying it (although it is possible to assign competencies to a person directly as well). With the increasing size of a company, the number of similar positions is likely to increase. In this case *position types* serve as a template for positions (e.g. “Secretary”) which can be refined in a position instance (e.g. “Secretary of Professor Becker”).

6.3. Substitute relationships

Not all workflow systems offer the possibilities of defining substitute relationships (proxies) between employees. In most cases the assignment of an activity to a role instead of a specific person solves the problem of substituting for an absent employee. In some cases (especially during an authorization or an inspection activity) it is necessary to assign an activity to a specific person. Substitutes are necessary in these cases in order to preserve the flexibility of the original role system. Most substitute mechanisms allow an actor to inherit *all* qualifications of the original actor. In some systems (e.g. WorkParty), the substitute mechanism relates two organizational positions instead of two actors. However, the complete inheritance of all existing rights can have serious impact on the security policies of an enterprise. Therefore it is desirable to be able to restrict a substitute relationship to specific roles of an actor. This is enabled by introducing a ternary relationship between the entity type person and the entity type role. A person can have several substitutes as well as be the substitute for several other people. The substitution may be reduced to certain roles of a person. Hence it is possible e.g. to create a substitute for a manager who is not allowed to sign orders over $ 50.000, because this specific role of the manager was excluded from the substitute relationship.

6.4. Process dependent staff resolution

During the instantiation of a workflow model it may be necessary to assign actors to activities depending on the state of process objects or the state of activities already performed. It should therefore be possible to assign an activity not only to the entity types of an organizational meta model, but also to the relevant relationship types, such as “supervisor of the current actor”. This option is useful in case of an exception, when an escalation mechanism redirects the current activity to the superior of the actor performing the activity. Another example of this mechanism is the double check of an invoice. In this situation it has to be guaranteed that the actor performing the second check is not the same actor who already performed the first check. For reasons of clarity, these relationships are not depicted in fig. 4.

An important aspect of process dependent staff resolution is the use of process object attributes as variables for the activity definition, which is outlined by means of an example. During the auditing of accounts, it is necessary to model separate activities for every origin of a foreign bill and to assign certain roles (Clerk with English/ Spanish language skills etc.) to each of these activities. With the use of process objects it is possible to reduce these activities to one, where the origin of the bill itself serves as a variable for the role definition, i.e. if an American bill is the process object of the current process instance, the generic role (“X language skills”) is replaced by a certain instance enhanced with the attributes of the process object (“English language skills”). Since this aspect affects the interface between the meta process model and the meta data model of a workflow management system, this aspect is not depicted in fig. 4.

6.5. Evaluation of individual meta models against a reference model

The comparison of the organizational meta models described in section 4.3. with the organizational reference model outlined above reveals some additional aspects. While the organizational meta model of WorkParty (fig. 2) shows entity types similar to those of the reference model (position type, organizational position etc.), the high number of relationship types tend to make the meta model unnecessarily complex. Contrary to this observation, the entity types of the FlowMark meta model (fig. 3) show only little resemblance to those of the reference model. On the other hand some relationship types may be addressed directly in FlowMark (coordinates, is member of), that are not possible in WorkParty.

The use of a reference model helps direct the attention of the observer to aspects that would not have been analyzed otherwise. Thus a reference model can improve the results of a meta model comparison. Reference meta models can 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.

7. Conclusions

The use of meta models for the evaluation of workflow management systems reveals some aspects, which would be difficult to discover using scoring models. The high degree of formalization and the proposed procedural model (conflict resolution, comparison of entity types, relationship types and attributes) can help potential workflow customers to determine the system that fits the business situation in the best way possible. If the business situation is depicted using formal methods (e.g. an organizational meta model) customers are able to formulate their demands for the ideal workflow system's features in a more precise way. This way, vendors, manufacturers and consultants are able to improve current systems and minimize the failure rate of workflow projects, which is still high at the moment. However, the modeling components that can very well be analyzed using meta models are - though important - not the only parts of a workflow system that have to be considered during a system evaluation. KOBIELUS points out the importance of finding the right system integrator as well as the stability and economic impact of a workflow system [18]. Technical and economic aspects of workflow management systems are evaluated best by the use of a scoring model. The selection of the right degree of detail as well as a preferred dimension for the comparison (cf. section 2) is of great significance for the success of such an approach. The specific requirements of the intended application area can be incorporated in two ways into the proposed methodology. The meta models of the systems analyzed can be opposed to company specific meta models which is feasible especially for the organizational meta models. Information about the business processes subject to workflow automation that relate to technical features of the systems analyzed should be incorporated into the scoring models. We are positive that a combination of scoring models and formal approaches for the evaluation of workflow management systems will enhance the customer's understanding of the business situation and the capabilities of the different workflow management systems. The empirical investigation of the proposed methodology based on a real world workflow project should be subject to future research.

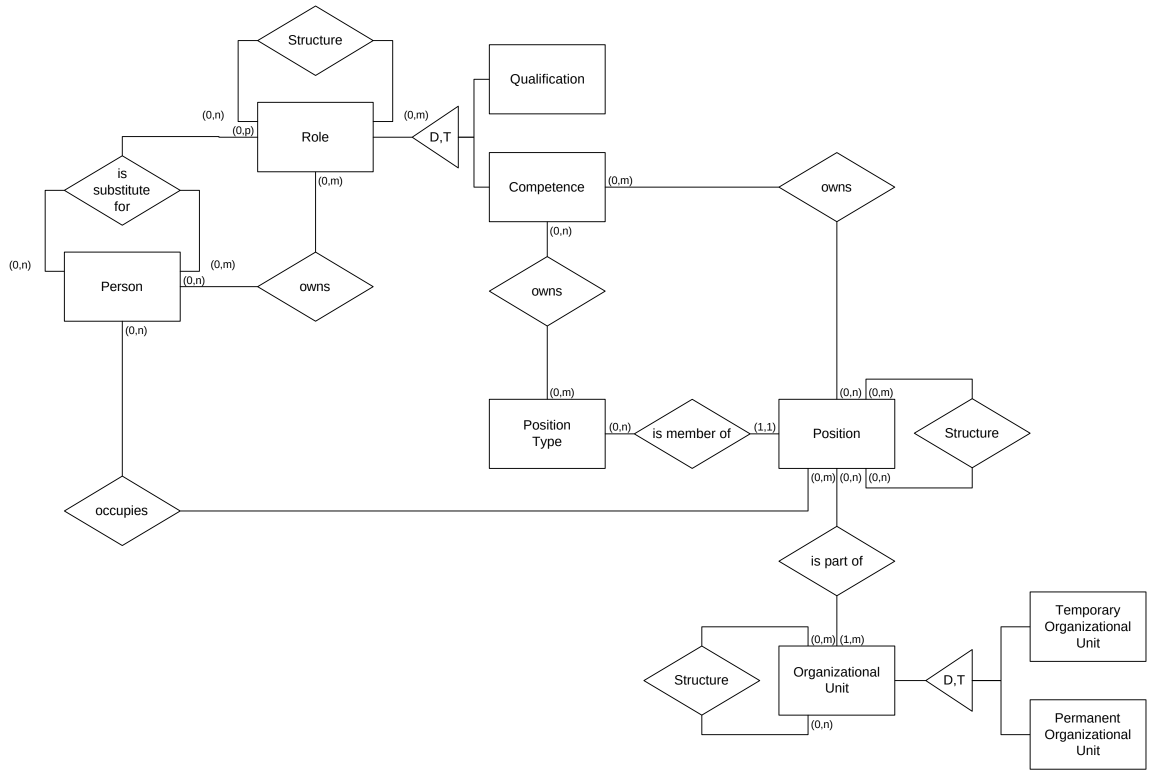

Figure 4. Organizational reference meta model

References

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

[2] Kazman, R., Abowd, G., Bass, L. and Clements, P.: Scenario-Based Analysis of Software Architecture. Technical Report GIT-CC-95-41. Georgia Institute of Technology. Atlanta 1995.

[3] Derungs, M.; Vogler, P.; Österle, H.: Kriterienkatalog Workflow-Systeme. Bericht Nr. 1 (IM HSG/CC PSI/1) des Kompetenzzentrums Prozeß- und Systemintegration. St. Gallen 1995. (in German)

[4] Camp, R. C.: Business Process Benchmarking - Finding and Implementing Best Practices. Milwaukee 1995.

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

[6] Heinrich, L. J., Damschik, I., Gappmaier, M. and Häntschel, I.: Workflow-Management. Produktevaluierung im Labor. In: HMD, 32 (1995) 181, pp. 101-112. (in German)

[7] Strahringer, S.: Konzept metamodellbasierter Methodenvergleiche. In: Schriften zur Quantitativen Betriebswirtschaftslehre, No. 2/96,. Ed.: W. Domschke, H. Stadtler, H. J. Petzold. Darmstadt 1996. (in German)

[8] Schwab, K.: Koordinationsmodelle und Softwarearchitekturen als Basis für die Auswahl und Spezialisierung von Workflow-Management-Systemen. In: Geschäftsprozeßmodellierung und Workflow-Management. Ed.: G. Vossen, J. Becker. Bonn et al. 1996, pp. 295-317. (in German)

[9] Rosemann, M. and zur Mühlen, M.: Evaluation of Workflow Management Systems - A Meta Model Approach. In: Proc. of the 2nd CAiSE '97/IFIP 8.1 Int. Workshop on Evaluation of Modeling Methods in SA&D. Ed.: K. Siau, Y. Wand, J. Parsons. Barcelona, June 16-17, 1997.

[10] Lei, Y. and Singh, M.P.: A Comparison of Workflow Metamodels. Proceedings of the ER-97 Workshop on Behavioral Modeling and Design Transformations: Issues and Opportunities in Conceptual Modeling, Los Angeles, November 1997.

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

[12] Steele, P. M. and 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.

[13] Saeki, M.: Object-Oriented Meta Modelling. In: Proceedings of the 14th Int. Conference on OO and ER Modelling. Ed.: M. P. Papazoglou. Berlin et al. 1995, pp. 250-259.

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

[15] Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen. Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung. Wiesbaden 1996. (in German)

[16] Batini, C. and Lenzerini, M.: A Methodology for Data Schema Integration in the Entity Relationship Model. IEEE Trans. on Software Engineering, 10 (1984) 6, pp. 650-664.

[17] zur Mühlen, M.: Der Lösungsbeitrag von Metamodellen und Kontrollflussprimitiven beim Vergleich von Workflowmanagementsytemen. Diploma Thesis. Department of Information Systems. University of Muenster 1996. (in German)

[18] Kobielus, J.G.: Workflow Strategies. IDG Books, Foster City, CA et al. 1997.