Using Conceptual Graphs for Organization Modeling in Workflow Management Systems

Jürgen Klarmann

Institute of Parallel and Distributed High-Performance-Systems (IPVR),
University of Stuttgart, Breitwiesenstr. 20-22,
D-70565 Stuttgart, Germany
klarmann@informatik.uni-stuttgart.de

Abstract. Workflow management systems reflect business structures in a separate organizational model and a process model because these models describe different aspects of business structures. The organizational model comprises members of an organization and organizational relationships (organizational structure) whereas the process model contains information about the tasks and their dependencies (process structure). Linking both of these models becomes more and more problematic because organizational models change much more nowadays than they have before and so lead to orphaned references in the process model. Moreover, organizational models should be unknown during process modeling, and equivalent process models should run with different organizational structures. For this, we introduce conceptual graphs for specifying organizational structures in the organizational model and for specifying potential task holders in the process model. We cover organizational knowledge and retrieve it at runtime exploiting the benefits of these knowledge representation technique. With this, it is possible to renounce name referencing and gain more degrees of freedom for the linking process.

1 Introduction

Workflow management systems reflect business structures in a separate organizational model and a process model because these models describe different aspects of business structures [1]. The organizational model comprises members of an organization and the organizational relationships (organizational structure) in order to be able to support the overall goal of an organization efficiently. The process model contains information about the tasks that have to be done, their dependencies (process structure) and the way how they are to be executed.

This separation aims to isolate changes in the organization from changes in the processes and vice versa. For example, the transfer of employees from one position in the organization to another does not need to be reflected in the process model, and a modification of a process part does not need to be reflected in the organizational model. Although both models cover independent aspects, they have to be linked up with each other because in the end, the activities has to be performed [2].

However, this act of linking becomes more and more problematic for two reasons:

1. Today, the enterprise environment is characterized in a stronger way by its turbulent behavior and its rapidly changing markets than some years ago [3, 4]. This tendency is caused by more demanding customer requests and economic conditions as well as social, moral and political conditions. For example, a company's division structure has to change more and more often to adapt to new market situations: scopes of duties are restructured and newly assigned to employees and obsolete ones are omitted. Changes in the organizational structure leave organizational models obsolete and cause orphaned references in the process model. This leads to inconsistent workflow specifications. Nowadays, the frequency of change is too high for being able to adapt the process models to the new organizational model manually. Nevertheless, former specified process models should still be executable under these circumstances to reduce the cost of changes.
2. Today, there is a tendency to more autonomous organization forms [5]. Company divisions have more rights in organizing their own structures. For example, a headquarter specifies a process that should be executed in different branches. Each branch has its own organizational structure that again is submitted to changes in the environment. This means that it is necessary to keep own process models for each branch organization even if the process specifies the same task structure. This handling is clumsy and not efficient. There is a need for supporting process executions on different organizational structures.

So far, existing approaches for organization modeling are based on singular concepts with pre-defined entities for modeling specific organizational structures. For supporting a more suitable linking between organizational structure and process structure it is necessary to cover organizational knowledge within the organizational model. We apply knowledge management techniques to achieve a more sophisticated linking process meeting the above aspects. We show how this can be done and how this knowledge-based approach for organization modeling can be used by exploiting the former implicit and now explicit knowledge. Additionally, a perpetual learning process improves the expressiveness of the knowledge base and so ensures a continuous corrective procedure within the knowledge management.

In section 2, we examine the problem domain of linking up between organizational model with process model in more detail and illustrate this with the help of a scenario. In section 3, we introduce conceptual graphs for specifying the organizational model as well as the task holder queries in the process model. Conceptual graphs are very suitable for knowledge representation. This knowledge is used for finding an eligible task holder. With this, linking succeeds even for changed or unknown organizational models. With the help of the scenario, we show a modeling as well as a linking and learning process. An assessment of related work is covered in section 4. We finish by giving a conclusion and an outlook in section 5.

2 Problem Domain

A process always runs against the background of a specific organizational structure. Traditional organizational models specify the organizational structure in terms of role, position, authority and organizational units [6–10]. According to specific organizational characteristics, existing organizational models differ in details for being able to express these specifics. So, various priorities of company are reflected in the definition and use of organizational elements and their interrelations. For instance, a basic approach for an existing traditional organizational models is shown in figure 1:

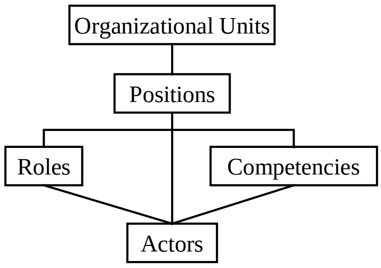


Fig. 1. Basic Organizational Model

The process model covers information about the tasks that have to be done, their dependencies and the way how they are to be solved. Nevertheless, information about the potential task holder specified in the organizational model has to be tackled, too [2]. This information is used for being able to assign tasks to a task holder.

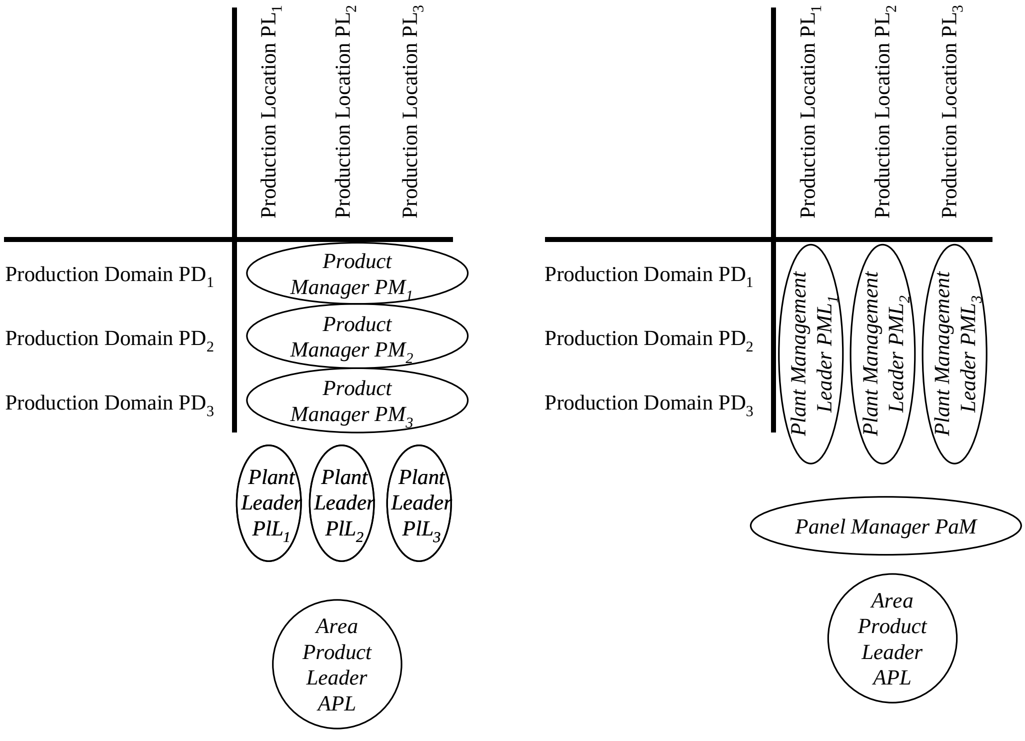
In already existing systems and research approaches, the relationship between the entities in the organizational model and the process model is hard-coded and very tight and is therefore not suitable to handle turbulent environments. The organizational model should not rely on specifying its entities in traditional singular terms of role, group, authority and organizational unit models as found in classical role models like [6, 10]. It should be possible to realize a more flexible detachment between the entities in the organizational model and those in the process model by leaving out pure name referencing and specifying knowledge about process execution in the organizational model.

Scenario

For illustrating this, we introduce the following scenario (figure 2): An enterprise should be restructured in the way that its branches can be managed more autonomously.

The enterprise has the production domains PD_1, PD_2 and PD_3, and the production locations PL_1, PL_2 und PL_3. For the sake of simplicity, we assume that each production domain is situated at each production location. So far, there has been a product manager who was responsible for a production domain, independent of its production locations. With this, rough planning results should have been put into action. Each plant of a production location is headed by a plant leader who is responsible for the operative execution and coordination of the production processes. The overall planning is taken over by an area product leader.

To reduce the coordination effort caused by orthogonal responsibility domains, the company decides to restructure itself in a way that emphasizes more strongly the autonomy of the several production locations. As a result of this, the production manager is abolished. Instead, the competency domain of the plant leader is extended by specifics about the production location (plant management leader). So, a leaner management with more room for autonomous manoeuvre is the result. Furthermore, a panel manager is installed who works in close cooperation with the different plant management leader for coordinating the autonomous branches.



a) Former Organizational Structure b) Changed Organizational Structure

Fig. 2. Restructuring of an Organization

Supporting Frequent Change

Restructurings like these have also to be reflected in the organizational model for keeping a specification up to date. Otherwise, the process models would not be executable any more because they would still contain references to former model elements, as for instance to the product manager (see figure 3). Existing workflow management models require a recheck of the process definitions and to adapt successively to the new references as panel management leader or panel manager in the restructured organizational model.

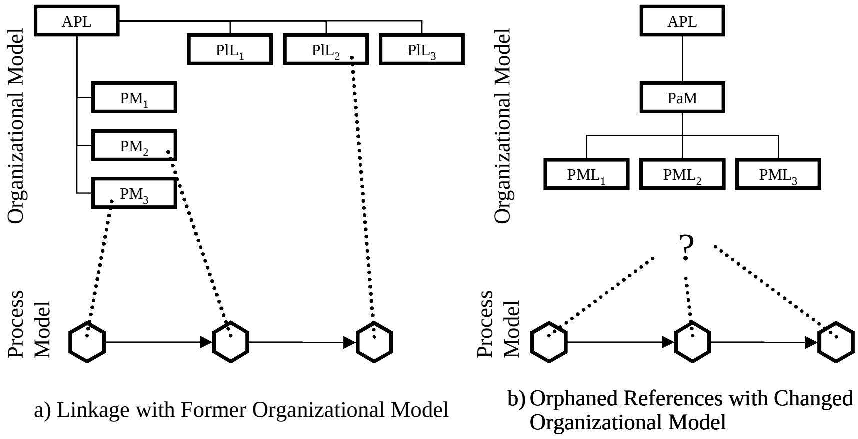


Fig. 3. Linking between Organizational Model and Process Model

The claim of a separate modeling of different aspects that has been originally intended by the separation of organizational model and process model is undermined because of hard-coded references. So, a proper independence of organizational model and process model is not given. This interweaving hampers an adequate behavior in turbulent markets. With change occurring ever so often, a manual adaptation of the process model cannot be done neither efficiently nor sufficiently.

Supporting Re-Use

The need for a stricter independency of organizational model and process model is also important regarding the global tendency towards self-organizing company branches. For instance, in the future a process should not be executed any more in the head office but instead in a division. However, the process model should be re-used and executed as usual because of cost and quality reasons. For example, the work has to be done in the head office by a product manager or by a plant management leader whereas in the division, it is done by a simple branch manager because of the smaller division size. It is the aim to be able to model process definitions that can be re-used and executed in different environments. Also in this case, the use of hard-coded references proves to be a handicap. In order to be able to execute a process model on different organizational models, linking up with each other must not be done by referencing organizational model elements but by specifying knowledge about possibilities of process execution in the organizational structure. Thus, the organizational model has to be able to cover more information than purely the name of organizational elements.

3 Organization Modeling

In the following, we introduce a knowledge-based approach that makes a redesign unnecessary after processes have changed and to allow for a re-use of process models for different organizational models. We propose an approach where the organization modeling in workflow management systems can be done independently of the process modeling. We achieve this by not considering the elements of the organizational model as atomic entities any more but instead as complex structures in an overall context. That means that we do not provide a pure name referencing. In our approach, the process model contains a content description of the executable tasks. So, potential task holders are defined by the characteristics of the task and, with this, are independent of a concrete organizational form.

In the organizational model, it is necessary to cover knowledge about the task characteristics that a potential task holder needs to possess for being able to take over that task. In principal, this knowledge is already present: For instance, a department chief knows from experience how to delegate tasks to his employees optimally concerning capacity, qualification, authority and so on. So far, this knowledge exists only implicit in his mind and is not explicitly specified. In some companies, it even exists as explicit profile specifications of employees, but it is not suitable for automatic workflow task assignment. It is necessary to develop an organizational model that is able to cover and retrieve explicitly this former implicit knowledge.

In order to link up the two models, the task characteristics described in the process model are compared to the described characteristics of the task holders in the organizational model. The description is specified in almost natural language format. We achieve independency between organizational model and process model by no more specifying the *who* of required or offered task holders. Instead, we specify the *what* that has to be performed.

If there is a counterpart in the organizational model to the characteristics described in the process model, the linking process succeeds. Otherwise, two cases can occur:

1. No eligible task holder can be found because there is no task holder that can fulfill the specified characteristics. This problem is system immanent and has to be solved ad hoc.
2. No eligible task holder can be found although there is someone who could fulfill the required characteristics. This case occurs because of the natural-language based specification of the task characteristics in the process model: Another specification was used than in the organization model.

For the second case, our approach offers the possibility of an evolutionary increase of the knowledge base: A continuous corrective procedure of knowledge acquisition (learning) can be initiated. With this, the task holder description included in the process model will be inserted in the organizational model and the query handling will succeed in future cases.

3.1 Organization Modeling with Conceptual Graphs

We use *conceptual graphs* for specifying task holder characteristics in the organizational model and task holder requirements in the process model. Conceptual graphs are a well-known method for knowledge representation [11]. They allow for a specification that follows human mind structures. They also have been developed to model the semantics of natural language. The semantics of a conceptual graph can be defined through mapping to a first order predicate formula. We use conceptual graphs for capturing knowledge about task holders in the organizational model as well as for specifying the request for a potential task holder in the process model.

We now briefly describe a conceptual graph. A more formal and detailed definition can be found in [12].

A conceptual graph is a finite, connected, directed bipartite graph. The nodes of the graph are either *concept* nodes or *relation* nodes. In our approach, a concept node represents an organizational element type. A relation node represents the kind of relationship that is established between concepts. A conceptual graph represents a type specification that stands in relation to other type specifications.

In our case, a concept node represents a concrete or abstract object of the organizational structure. Referring to the introduced scenario, an example for a concrete concept is a task holder or a production location including derived instances (a concrete person as instance for the concept task holder and PL_1 as instance for the concept production location). An example for an abstract concept is a plant management leader or a production domain with their instances PML_1 and PD_1 that have no physical equivalent. A relation expresses a specific relationship between organizational elements. In the visual representations, concept nodes are specified in the following as rectangles (alternatively surrounded by square brackets) whereas relation nodes are pictured as round ellipses. Concept instances are written within braces.

To illustrate the visual representation of a conceptual graph in our case, we describe the changed organizational model of the scenario mentioned above in figure 4. The natural language interpretation of this graph can be formulated as “A plant management leader called PML_3 is responsible for the production domains PD_1, PD_2, and PD_3 as well as for the production location PL_3. He is a member of panel P. Panel Manager PaM is head of panel P whereas the area product leader APL is superior to PaM.” This is explained by the concepts “PRODUCTION DOMAIN”, “PRODUCTION LOCATION”, “PLANT MANAGEMENT LEADER”, “PANEL”, “PANEL MANAGER” and “AREA PRODUCT LEADER” and by the relations “IS RESPONSIBLE”, “IS MEMBER OF”, “IS HEAD OF” and “IS SUPERIOR TO”. With this, it is possible to refer to a plant management leader not only by declaring his name but also by explaining in terms of his environment.

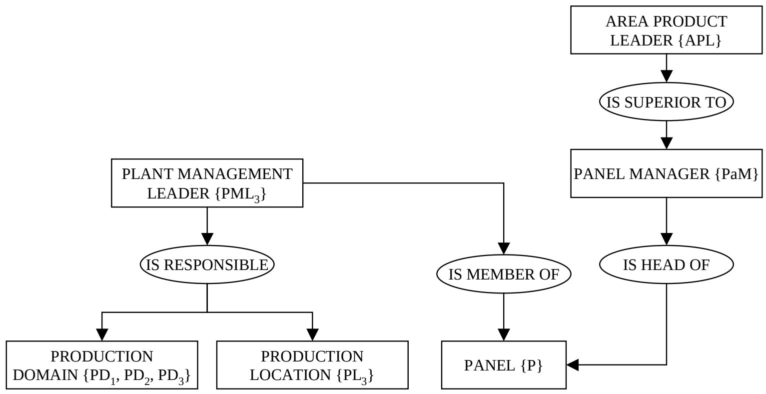


Fig. 4. Visual Representation of a Conceptual Graph Describing a Plant Management Leader

A concept node contains two parts: a concept type and a list of instances. For instance, the concept type of the concept "[PLANT MANAGEMENT LEADER]" in figure 4 is "PLANT MANAGEMENT LEADER" and has the instance "PML_3". In our approach, we use concept types for modeling the different organizational structure types.

All organizational elements of other approaches such as role, position, competency and even authority can be described as concept types. Also, new organizational structure types can be defined. With this, we embrace the expressiveness of all existing approaches. Additionally, it is possible to specify any connection types between organizational elements.

For achieving a platform of a common understanding, concepts and relations have to be well-defined. The use of ontologies for describing important concepts and the relationships between these concepts is therefore essential [13]. For instance, [14] presents a generally applicable ontology of business enterprises. It can be seen as a base for our specification with conceptual graphs and can be used for determining a basic vocabulary with a well-defined meaning for specifying the organizational model.

Concepts are set in relation to each other within a concept hierarchy. This hierarchy expresses specialization. "SOMETHING" is a pre-defined type that is at the top of any concept hierarchy. So, any concept type is derivable. For more details concerning conceptual graphs ontologies, see [15].

In our approach, we can use it to express compatibilities between organizational structure types. The concept hierarchy also contains concepts that are not part of an actual organization form anymore. Because of that, obsolete requests can be handled. These concept nodes are equal to the other ones, they differ only in not having instances. They serve as entrance points for navigating in the concept hierarchy.

Analogously to the concept hierarchy, there is a relation hierarchy that allows for specifying specializations between relations. This aspect will not be discussed here in more detail.

Figure 5 shows an excerpt of the concept hierarchy for a production scenario:

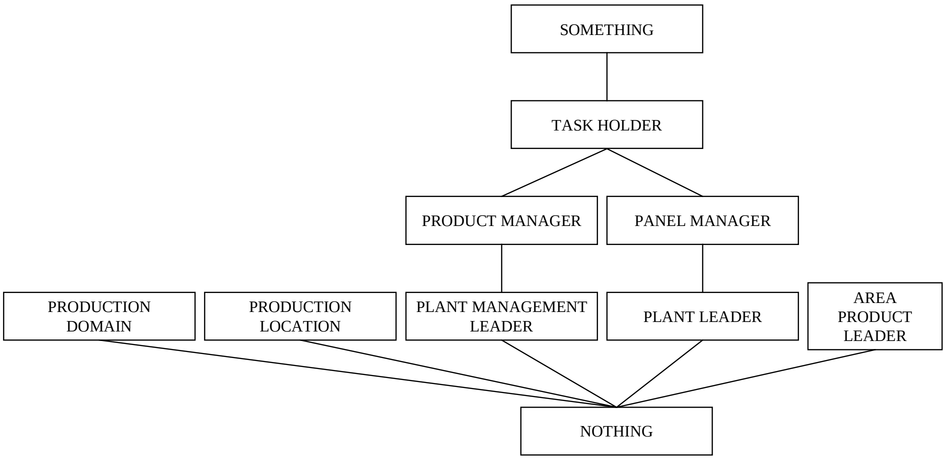


Fig. 5. Concept Hierarchy Describing Specialization Relations in the Organization

Whereas in the organizational model, conceptual graphs are used to specify task holder characteristics, in the process model, they are used to specify a query for a required task holder.

To illustrate this, we consider once more our scenario: In the process model, it is searched for potential task holders for several production domains and production locations. One of such specifications could be "a task holder who is responsible for production domain PD_1 and for production location PL_3". This statement can be translated into a conceptual graph whose visual representation is shown in figure 6. The symbol "*" denotes a generic object that will match with any other object. For more specification details of conceptual graphs, see [12].

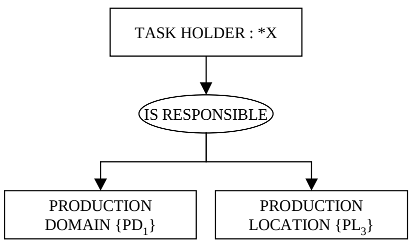


Fig. 6. A Conceptual Graph Picturing a Query

3.2 Linking Policy with Conceptual Graphs

After having presented the organization modeling with help of conceptual graphs, we show how this can be used for assigning task holders by exploiting the covered knowledge.

For matching an offered conceptual graph and a required conceptual graph, a type conformity between conceptual graphs has to be defined [16]. In brief, two conceptual graphs are type conform if their node sets and relation sets can be mapped injectively to each other while keeping their structures. Extended type conformity uses additional information about type hierarchy and relation hierarchy.

To illustrate type conformity, we reconsider our scenario again. The query conceptual graph depicted in figure 6 is compared to the conceptual graph of the organizational model depicted in figure 4. This query will match because the query conceptual graph is extended type conform to the conceptual graph of the knowledge-base. That means that they can be mapped injectively to each other while keeping their structures when we consider that the type “PLANT MANAGEMENT LEADER” is a specialization of “TASK HOLDER”. With this, the product management leader PML_3 can be found as task holder.

Conceptual graphs are not only useful for expressing the characteristics of potential task holders; they can support a correct assignment even if obsolete organizational model elements are used. For instance, a further process model contains a query for an obsolete “product manager (who is responsible for production domain PD_1 and for production location PL_3)”. Since the concept type “PLANT MANAGEMENT LEADER” is a specialization of the concept type “PRODUCT MANAGER”, extended type conformity is given and the plant management leader PML_3 can be found. That way, we achieve an immense expressiveness: No corset in form of a restricting meta model dictates the use of predefined model elements. The assignment at runtime succeeds because knowledge about the task descriptions is sufficiently exploited. The underlying ontology with its well-defined vocabulary and meaning [14] ensures that the same characteristics are specified in both the organizational model and the process model.

As an example for a learning process let us consider the query for “a task holder who coordinates the production location PL_3”. Figure 7 shows the visual representation of the corresponding conceptual graph. We assume that “COORDINATES” is a pre-defined relation.

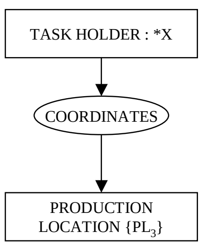


Fig. 7. A Further Conceptual Graph Picturing a Query

This query cannot be resolved because there is no extended type conformity between the conceptual graph picturing a query and the conceptual graph representing the organizational structure. Intuitively, it seems clear that only the Panel Manager PaM could be assigned for this task. He is not found because this aspect has not been covered in the knowledge base so far. If it is consensus, that this has been a query for Panel Manager PaM, this information can be added to the organizational model. For that, the query conceptual graph is inserted in the conceptual graph representing the organizational structure. The visual presentation of the extended conceptual graph is shown in figure 8. After having made this extension the query will match next time.

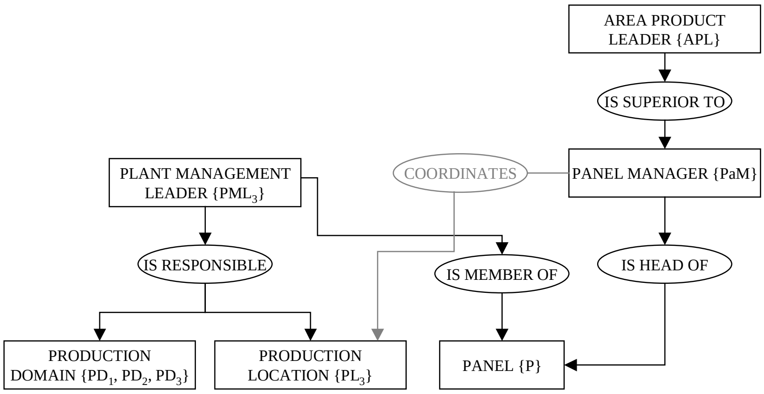


Fig. 8. Enhanced Conceptual Graph after Having Learned

4 Related Work

It is generally accepted that there is a need for providing a flexible execution mechanism in the domain of workflow management [17]. Besides the need to support change in the organizational model that this paper deals with, there is a crucial demand for being able to support the change of workflow models and to allow for flexible control and data flow. Work which supports workflow models uses versioning to control changes and to ensure a consistent propagation of model changes to running instances [18, 19]. Concerning flexible control and data flow execution, descriptive modeling constructs like in [20], and flexible execution mechanisms like late binding of subworkflows [21] have been developed on the one hand. On the other hand, controlled adaptations of running workflow instances have been examined [22].

Research on change of the organizational models has revealed that traditional workflow management systems (for instance InConcert [23] and COSA [24]) are not capable of addressing the requirements of an organization independent linking. Their assignment of potential task holders to processes is based on referencing names of organizational elements that hampers a more flexible coupling between organizational model and process model. [25] provide a more flexible assignment between organizational model and process model in specifying agents that return a reference to potential task holders. With this, it is possible to express more complex assignment rules. Nevertheless, their approach is based on name referencing and does not cover organizational information.

Work in modeling the organizational aspect is done by [7, 8]. They extend in their organization meta model the basic approach in [6] by introducing further organizational elements. [8] introduces new elements as “organization resource” whereas [7] includes explicitly the substitute rule in his meta model. This allows them to define a more detailed organizational model covering more aspects which results in more suitable assignments. Their meta models allow to derive all organizational elements that they have foreseen. This is useful for many cases but not generic enough to define any organizational structures.

[26] has recognized that there is not only the need for a powerful organizational model but also for a powerful meta model with which to arrive at more flexibility in organization modeling. He introduces a generic meta model that leaves out organizational specific elements and so allows to define any organizational model. His focus lies on generality and less on the treatment of structural change.

Similar to our approach, [27–29] allow for more flexibility in the workflow domain by using knowledge-based techniques. Organizational knowledge is covered in order to provide an improved decision support during different process stages. Hence, their approaches are mainly based on the use of ontologies [14, 30, 31]. They are able to describe precisely relevant issues with their interrelations and gain a common understanding. As we do, they exploit the additionally captured knowledge for making process mapping more sophisticated. The key feature of these approaches is the use of a well-defined specification that includes a common understanding. Though they do not offer the possibility to outline task holders in a query expressed in everyday language neither they present a way to handle expressions which are so far unknown. Furthermore, simple ontologies do not offer the resolution possibilities as conceptual graphs do because of their closeness to predicate calculus [11].

Versioning approaches [32] deal with organizational restructurings as well. They adapt assignment policies to the changed organizational model by focusing on consistency preservation. Likewise, they cover organizational information in the process model and are based on name referencing.

5 Conclusion and Outlook

In this paper we have introduced conceptual graphs for specifying task holder characteristics in the organizational model and task holder requirements in the process model. With this, organization modeling in workflow management systems can be performed independently of process modeling.

We have motivated that there is a need for a stricter detachment between organizational model and process model: On the one hand, organizational structures change more and more often than they did before. After having changed they leave orphaned references in the process model because they refer to organizational elements by names. Nowadays, the frequency of change is too high for being able to adapt process models to the new organizational model manually. On the other hand, the tendency towards self-organizing company branches increases. Organization branches can build up and handle organizational structures more and more autonomously. Because of name references to organizational elements in the process models, each branch requires to keep an own process model even if the process specifies the same task structure. This becomes very inefficient; instead, a process model should be able to be executed in different organization branches. Generally, we have seen that name referencing hampers the linking between process model and organizational model and there is a need for a stricter detachment and an alternative linking method.

We have shown how we can achieve this by using conceptual graphs. With conceptual graphs, we are able to make intuitive descriptions in a natural-language based way and to consider organizational model elements not as atomic entities any more but instead as complex structures embedded in an overall context. In the organizational model, we use this for specifying a knowledge base that covers information about the characteristics of the task holders. In the process model, we use this for specifying a query for a potential task holder by describing the task requirements rather than simply the task holder's name. With this, we achieve independency of a concrete organization form.

We have presented how our approach enables the assignment between organizational model and process models. Furthermore, we have presented, how a continuous corrective procedure can occur. We have described some scenarios and have shown how we are able to deal with the identified requirements by applying the approach to the introduced scenarios.

To demonstrate our concepts, we are about to develop a prototype that uses conceptual graphs for the specification of the organizational model. In order to follow up this aim, we have to extend the commercial workflow management system HP Changengine [33] by our own organizational component. This allows us to describe the organizational elements in the way we have outlined in the course of this paper and to adapt the assignment component. Changengine itself offers certain interfaces which are appropriate for this purpose.

Because we cannot directly specify roles via conceptual graphs in Changengine, we use the names of these roles as identifiers for the corresponding graphs, which themselves will be stored in a suitable database. As a straightforward means for generating these non-ambiguous identifiers we use the linear notation of conceptual graphs [11]. By proceeding like this, the extended assignment component will be able to resolve the role name and to match the requested conceptual graph with the one provided.

References

1. S. Jablonski, C. Bußler: Workflow Management. Modeling Concepts, Architecture and Implementation. International Thomson Computer Press, September 1996.
2. C. Bußler, S. Jablonski: An Approach to Integrate Workflow Modeling and Organization Modeling in an Enterprise. In: Proceedings of the 3rd IEEE workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE), Morgantown, West Virginia, USA, 1994.
3. R. Wigand, A. Picot, R. Reichwald: Information, Organization and Management: Expanding Markets and Corporate Boundaries. John Wiley & Sons, Ltd., 1997.
4. E. Westkämper, H.-H. Wiendahl, P. Balve: Dezentralisierung und Autonomie in der Produktion (in German). In: ZWF 93 (1998) 9, Hanser-Verlag, 1998.
5. G. Reinhart: Vom Wandel der Zeit: Wandel als Chance für unsere Unternehmen im globalen Wettbewerb (in German). In: ZWF 94 (1999) 1-2, Hanser-Verlag, 1999.
6. W. Rupietta: Organization and Role Models for Workflow Processes. In: Peter Lawrence (Eds.): Workflow Handbook 1997, John Wiley & Sons Ltd, 1997.
7. J. Galler: Metamodelle des Workflow-Managements (in German). Veröffentlichung des Instituts für Wirtschaftsinformatik, Heft 121, Universität des Saarlandes, 1995.
8. H. Heilmann: Die Integration der Aufbauorganisation in Workflow-Management-Systeme: In: H. Heilmann, L. Roithmayr (Eds.): Information Engineering. Wirtschaftsinformatik im Schnittpunkt von Wirtschafts-, Sozial- und Ingenieurswissenschaften (in German), R. Oldenbourg Verlag München, Wien, 1996.
9. CommonKADS W. Post, B. Wielinga, R. de Hoog, G. Schreiber: Organizational Modeling in CommonKADS: The Emergency Medical Service. In: IEEE Expert 12(6), 1997.
10. R. Hoog, B. Benus, C. Metselaar, M. Vogler, W. Menezes: Organisation Model: Model Definition Document. Technical Report, University of Amsterdam, KAD-SII/M6/UvA/041/3.0, Netherlands, 1994.
11. J. F. Sowa: Conceptual Structures, information processing mind and machine. Addison-Wesley Publishing Company, 1984.
12. J. F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, USA, 2000.
13. M. Uschold, M. Gruninger: Ontologies: Principles, Methods and Applications, The Knowledge Engineering Review, 13(1), 1998.
14. M. Uschold, M. King, S. Moralee, Y. Zorgios: The Enterprise Ontology. The Knowledge Engineering Review, Vol. 13, Special Issue on Putting Ontologies to Use (Eds. Mike Uschold and Austin Tate), 1998.
15. A. Puder, S. Markwitz, F. Gudermann, K. Geihs: AI-based Trading in Open Distributed Environments. In 3rd International IFIP TC6 Conference on Open Distributed Processing (ICODP'95), Brisbane, Australia. Chapman and Hall. 20-24 February 1995.
16. A. Puder: Type systems for the trading of services in open distributed environments. (in German) Ph.D. Thesis, University of Frankfurt, 1996/97.
17. P. Heinl, S. Horn, S. Jablonski, J. Neeb, K. Stein, M. Teschke: A Comprehensive Approach to Flexibility in Workflow Management Systems. In: Proceedings of the ACM 1999 Conference on Work Activities Coordination and Collaboration (WACC'99), San Francisco, California, USA, 1999.
18. M. Kradolfer, A. Geppert: Dynamic Workflow Schema Evolution Based on Workflow Type Versioning and Workflow Migration. In: Proceedings of the Fourth International Conference on Cooperative Information Systems CoopIS'99, Edinburgh, Scotland, September 1999.

19. G. Joeris, O. Herzog: Managing Evolving Workflow Specifications. In: Proceedings of the Third International Conference on Cooperative Information Systems CoopIS'98, New York, USA, August 1998.
20. S. Jablonski: MOBILE: A Modular Workflow Model and Architecture. Proceedings of the Fourth International Working conference on Dynamic Modelling and Information Systems, Noordwijkerhout, Netherlands, September 1994.
21. J. Hagemeyer, T. Herrmann, K. Just-Hahn; R. Striemer: Flexibilität bei Workflow-Management-Systemen (in German). In: Software-Ergonomie '97: Usability Engineering: Integration von Mensch - Computer - Interaktion und Software-Entwicklung. Stuttgart, Teubner-Verlag, 1997.
22. R. Siebert: An Open Architecture For Adaptive Workflow Management Systems. In: Transactions of the SDPS: Journal of Integrated Design and Process Science. Society for Design and Process Science, Austin, Texas, USA, 1999.
23. TIBCO: Enterprise Business Integration With TIB/InConcert. A Technical Overview. Feb. 1999.
24. LEY: COSA Workflow 2.0 Product Specification, 1999.
25. C. Bußler, S. Jablonski : Policy Resolution for Workflow Management Systems. In: Proceedings of the Hawaii International Conference on System Sciences 28, Maui, Hawaii, January 1995.
26. C. Bußler: Organisationsverwaltung in Workflow-Management-Systemen (in German). Deutsche Universitäts-Verlag, 1998.
27. B. Dellen, F. Maurer, G. Pews: Knowledge- based Techniques to Increase the Flexibility of Workflow Management. Data and Knowledge Engineering, North-Holland, 1997.
28. J. Stader, J. Moore, P. Chung, I. McBriar, M. Ravinranathan, A. Macintosh: Applying Intelligent Workflow Management in the Chemicals Industries. In: Layna Fischer (Eds.): Workflow Handbook 2001, Future Strategies Inc., 2000.
29. P. Jarvis, J. Stader, A. Macintosh, J. Moore, and P. Chung. 1999: What Right Do You Have to Do That? Infusing Adaptive Workflow Technology with Knowledge about the Organisational and Authority Context of a Task. In: Proceedings of the First International Conference on Enterprise Information Systems (ICEIS-99), Setubal, Portugal, 1999.
30. J. Stader, A. Macintosh: Capability Modelling and Knowledge Management. In: Applications and Innovations in Expert Systems VII, Proceedings of ES 99 the 19th International Conference of the BCS Specialist Group on Knowledge-Based Systems and Applied Artificial Intelligence, Cambridge, December, 1999.
31. J. Moore, J. Stader, P. Chung, P. Jarvis, A. Macintosh: Ontologies to Support the Management of New Product Development in the Chemical Process Industries. In: Proceedings of the International Conference on Engineering Design (ICED 99), Munich, Germany, 1999.
32. C. Bußler, S. Jablonski: Datenbankunterstützung für Workflow Management (in German). In: S. Jablonski (Eds.): Database Support for open Workflow Management Systems. Arbeitsbericht des Instituts für mathematische Maschinen und Datenverarbeitung (Informatik), Universität Erlangen, Nr. 5, Band 29, Mai 1996.
33. HP Changengine, Technical Reference Guide. Edition 4.5, 2000.