zur Muehlen, Michael; Becker, Jörg: WPDL – State-of-the-Art and Directions of a Meta-Language for Workflow Processes. In: Bading, L. et al. (Ed.): Proceedings of the 1st KnowTech Forum, September 17th-19th 1999, Potsdam 1999.

Workflow Process Definition Language – Development and Directions of a Meta-Language for Workflow Processes

Michael zur Muehlen, Jörg Becker

University of Muenster
Department of Information Systems
Steinfurter Str. 109
48149 Muenster
Phone +49 (251) 8338080
Fax +49 (251) 8328080
E-mail {ismizu|isjobe}@wi.uni-muenster.de

Abstract

Meta-Languages for the definition of processes serve several purposes. They can be employed as an integration platform for the exchange of process models that are specified in proprietary languages, their expressiveness can serve as a benchmark for the selection of a application specific modeling language and they can be used for the application-independent specification of process models that can then be transformed into the language relevant for the domain-specific context. In this paper we outline several approaches of meta-languages for process specification and compare them to the Workflow Process Definition Language as defined by the Workflow Management Coalition.

1 Meta-Languages for Workflow and Process Modeling

1.1 Workflow Management Technology

The support of process enactment through workflow management systems has increased significantly in both administrative and technical domains. The development of workflow technology can be traced back to various origins, such as office information systems [1], computer supported cooperative work (CSCW) [2], imaging and document management [3] as well as advanced database technologies [4, 5] (for a discussion of the origins of workflow cf. e. g. [6]). Workflow management systems support the enactment of processes by coordinating the temporal and logical order of the elementary process activities (sometimes called elementary workflows and described as the behavioral and functional/control-flow aspect of a workflow) and supplying the data, resources and application systems necessary for the execution of the functions (these are the informational, organizational and operational/technological aspects of a workflow, for a discussion of the different aspects cf. [7]). Most workflow management systems distinguish between a buildtime component, used for the specification of workflow models, and a runtime component that is used during the invocation of workflow instances. The buildtime models are either interpreted by the runtime engine or compiled into a pseudo-code that can be executed by the runtime component. In the recent past several research projects focused on the weakening of this separation, thus enabling the modification of running process instances or the ad-hoc planning of process parts that are unknown at buildtime (cf. e. g. [8, 9]).

1.2 Languages for Workflow Process Specification

For the specification of workflow processes, several languages have evolved over time. Almost any vendor relies on a proprietary format for the specification of workflow models, only few use existing process modeling languages such as high-level Petri-Nets for the modeling component of their product (cf. e. g. the discontinued LEU system, that used FUNSOFT-nets for the specification of workflow processes [10]). According to CARLSEN, the existing languages for workflow process modeling can be classified in five distinct groups [9]:

- IPO (Input-Process-Output)-based languages, such as the activity networks used in IBM MQSeries Workflow [11]. These languages describe a workflow as a directed graph of activities, denoting the sequence of their execution.
- Speech-Act-based approaches (sometimes called Language Action approaches) as used in Action Technologies Action Workflow product [12]. These approaches model a workflow as an interaction between (at least) two participants that follow a structured cycle of conversation. Namely the phases negotiation, acceptance, performance and review are distinguished.
- Constraint-based modeling methods, such as Generalized Process Structure Grammar (GPSG), proposed by GLANCE et al. [13]. These approaches describe a process as a set of constraints, leaving room for flexibility that is otherwise governed by the restrictions of the IPO- or Speech-Act-based approaches. Constraint-based modeling languages are typically text-based and resemble traditional programming languages, whereas a graphical representation of these models seems difficult.
- Role-modeling based process descriptions, such as Role Activity Diagrams (RADs).
- Systems thinking and system dynamics, that are used in conjunction with the concept of learning organizations (cf. e. g. [14]).

In the following sections we will focus on the first three classes of modeling languages, namely IPO-based, Speech-Act-based and Constraint-based languages.

1.3 Meta-Languages for Process Specification

The diversity of approaches and tools as described in the previous sections leads to severe problems for users, that wish to employ several products that follow different modeling paradigms, as well as for users that plan to integrate their business processes with previous and latter stages of the value chain, i. e. suppliers and customers. In order to facilitate the integration of different tools from different vendors, several initiatives have emerged that focus on the development of meta-languages for process modeling.

A meta language is a superset of constructs that can be found in process modeling languages and that can be used to map concepts from one process modeling language through a construct of the meta language to a related concept of another modeling language.

1.4 Related Work

In the field of database management systems the concept of schema integration has been extensively discussed in the literature (cf. e. g. [15]), however, in the field of process management the close relationship between the syntax and the semantics of a process model create new problems.

KNUTILLA et al. present a comprehensive evaluation of process modeling languages [16]. In the course of the NIST PSL project (see section 2.3) twenty-six process specification languages were evaluated with regard to their applicability to the manufacturing domain, which serves as the origin for the PSL requirements.

The evaluation of the modeling methods of workflow management systems can be found in two approaches. HEINRICH et al. focus on the practical effects of workflow technology in a laboratory study, comparing process enactment with and without workflow support [17]. LEI and SINGH [18] as well as ZUR MÜHLEN [19] focus on the evaluation of workflow management systems using their meta models. However, the grammar of the underlying modeling methods is not analyzed in detail.

2 Five Meta-Languages for Process Modeling

In the following sections we present five distinct approaches of meta languages for process modeling: Workflow Process Definition Language, Process Interchange Format, Process Specification Language, Generalized Process Structure Grammar and the Unified Modeling Language. For each of these languages their origin and current status are described.

2.1 Workflow Process Definition Language (WPDL)

The Workflow Process Definition Language (WPDL) was first introduced by Workflow Management Coalition (WfMC) in 1994. The WfMC is a non-profit, international organization committed to the development of standards in the fields of workflow technology. The main work of the WfMC focuses on the standardization of terms used in the context of workflow management applications [20] and the interoperability of different systems [IOP]. Currently the WfMC consists of over 200 members, namely vendors, users, research institutions and analysts dealing with workflow technology. The WfMC consists of two main groups: The technical committee (TC), which is organized in working groups that address various areas of workflow technology and develop standard documents, and the external relations committee (ERC), which is responsible for the publication of the TC standards and consists of a number of country contacts for parties interested in the work of the WfMC. Both groups are controlled by a steering committee that supervises the consistency of the work and sets the general direction for the work of the WfMC. Although a standardization body in its own right, the WfMC works closely together with other standardization organizations, such as the Object Management Group, whose workflow management facility [21] was submitted by a consortium of WfMC member companies.

The specification of the WfMC form around a reference model (cf. fig. 1) that describes the basic elements of a workflow management system with a focus on the interfaces to external systems. This reference model should not be confused with the internal representation of a workflow management system as described e. g. by JABLONSKI and BUSSLER [7], since the model in figure 1 only describes those parts of the workflow management system that are potential candidates for interoperability with other software systems.

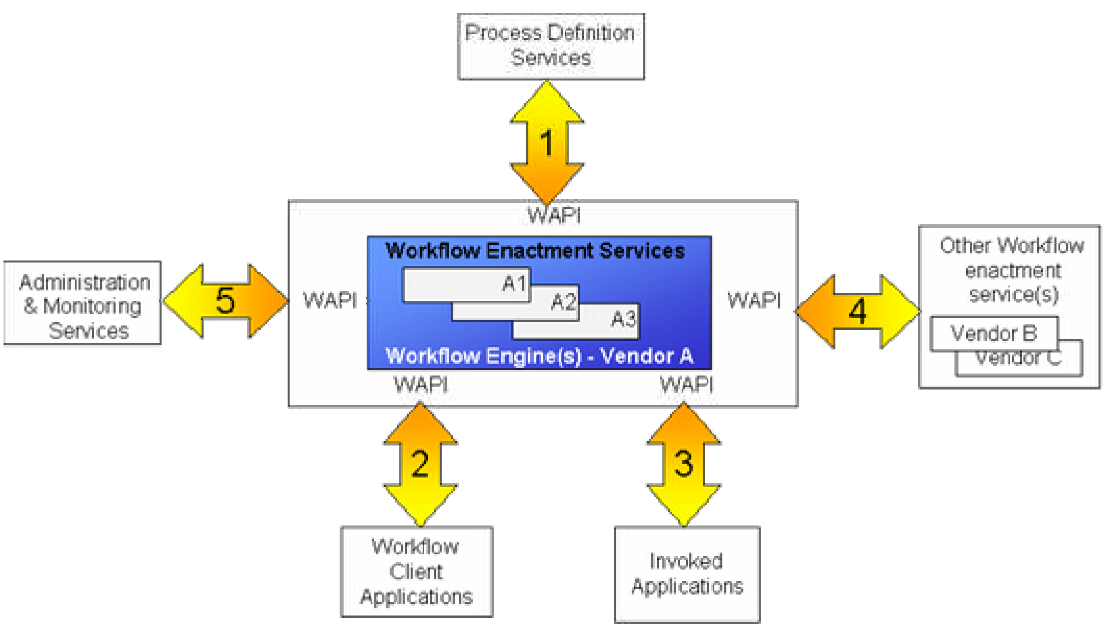

Figure 1. Workflow Management Coalition Reference Model

The five interfaces of the reference model are linked to the workflow enactment services via a Workflow Application Programming Interface (WAPI), that has been defined on an abstract level [22]. The relevant interface for the exchange of process model data is Interface 1 [23]. The Workflow Process Definition Language (WPDL) was established as a meta-language for the exchange of buildtime workflow process models through a batch procedure (import/export of process models). The keywords of WPDL are based on the terms defined in the WfMC glossary. The language design is based on a minimum meta-model that defines the elementary components that have to be supported by a tool that reads and/or writes WPDL. This minimum meta-model can be extended by vendor-specific extensions.

The meta-model of the process part of WPDL is depicted in figure 2 (for reasons of simplicity the cardinalities of the relationship types have been left out of the diagram). The organizational/resource model of WPDL is a specialization of the Workflow Participant Specification Entity Type and thus not displayed in the figure. Core concept of the WPDL is a Workflow Process Definition that is composed of one or many Workflow Process Activities. The ordering of activities is determined by Transition Information elements that connect single activities. For more complex routings a Transition may rely on Workflow Relevant Data, that is, data from application systems which is relevant for the sequence of activities (i. e. the amount of a credit request that determines, whether the VP of the bank has to sign the approval or not).

The entity types of WPDL are not extendable, however, user-defined attributes may be added to the single entity types. Moreover, references to external data sources as connecting points are explicitly denoted, such as the referral to an external organizational repository, to system and environmental data or to invoked application systems.

In order to establish conformance for workflow management systems that follow different modeling paradigms, several conformance classes have been defined. These conformance classes limit the number of elements a workflow management system has to support in order to claim conformance to WPDL. These restrictions include e. g. a block-restriction for workflow systems, that require each split in the process to be followed by a similar join in a later part of the process.

Currently a mapping between WPDL and XML is under discussion in order to enable the exchange of process models through an internationally defined standard encoding language. A standard for this mapping is expected to be released within the year 2000.

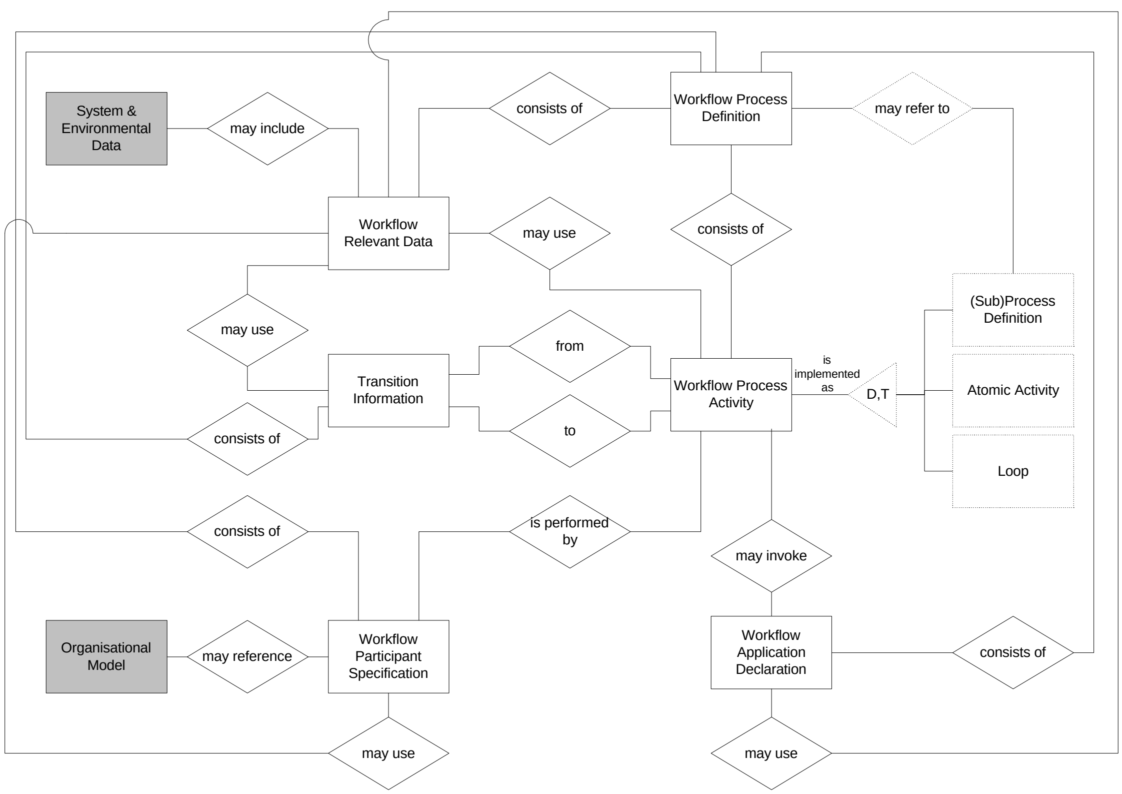

Figure 2. Meta model of the WPDL process modeling elements.

2.2 Process Interchange Framework (PIF)

The Process Interchange Framework was developed as a standardized language for the processes recorded in the MIT Process Handbook project [24]. The Process Handbook project is targeted at the collection of representative business processes from different organizations and the presentation of these processes in order to facilitate the comparison and selection of alternative processes in actual business situations. Its main purpose is the support of organizations seeking the redesign of existing processes and the support of new processes that emerge due to technological support.

Within the PIF approach, processes are represented at various levels of abstraction, derived from the object-oriented concept of inheritance and dependency management as in coordination theory. The creators of PIF describe the main advantage of the concept as “it allows users to explicitly represent the similarities (and differences) among related processes and to easily find or generate sensible alternatives for how a given process could be performed.” [25]

All constructs of the PIF Core are specified in the Knowledge Interchange Format (KIF), a language that is designed for the interchange of knowledge among separate computer systems [26]. KIF allows for an extension of existing concepts, which is important for the adding of user-defined extensions to the PIF language core. Furthermore, KIF is a proposed standard and has well-defined formal semantics, that simplify the process of defining the core PIF constructs.

A process description in PIF is based on a set of frame definitions. Each of these frame definitions denotes an entity type that can be instantiated (for example TIMEPOINT or ACTIVITY), these types are arranged in a hierarchy. The hierarchy of PIF core components is depicted in figure 3. For each type in PIF there exists a set of predefined attributes which define various aspects of that instance of this type. As an example, the CREATES definition has an ACTIVITY and an OBJECT attribute, the values of which give the object(s) being created and the activity which creates the object(s). Attributes are inherited from supertypes to types as well from types to their instances. An instance of the ACTIVITY frame definition for example contains the attribute Name because the type ACTIVITY inherited this attribute from its supertype ENTITY. The value of an attribute within one frame may refer to another frame. This way relationships between the instances of these frames can be represented.

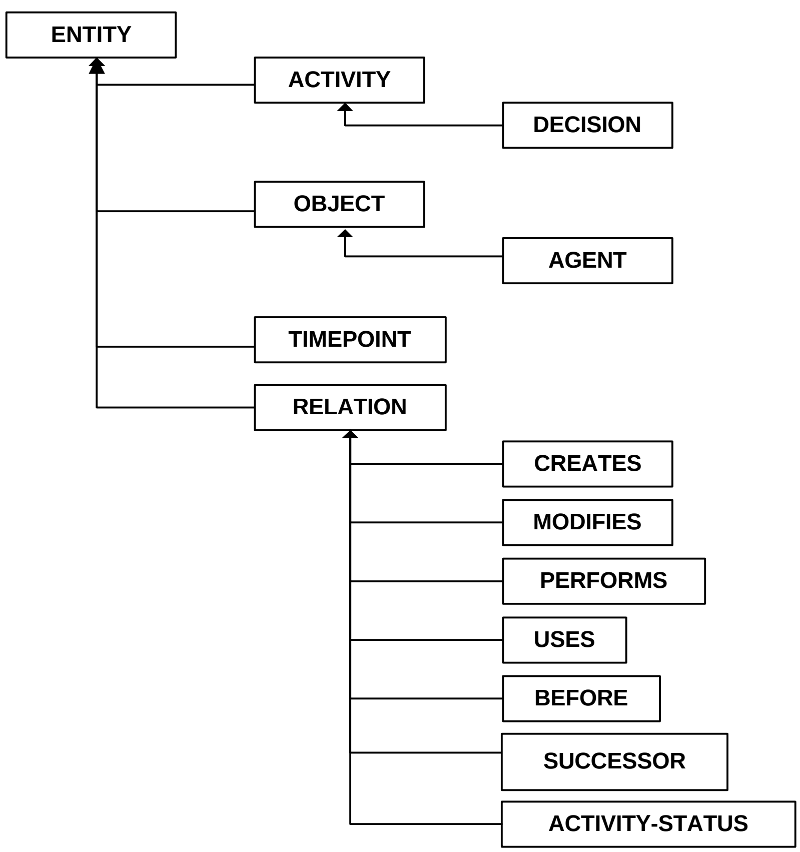

Figure 3. Hierarchy of PIF core components (Source: [PIF])

The Process Interchange Framework is a powerful exchange platform for process models. Due to its modular design it can easily be extended to accommodate the needs of workflow process modeling.

The PIF working group is exchanging ideas with the Workflow Management Coalition about making PIF and WPDL interoperable, thus paving the way for a unified interchange format.

2.3 Process Specification Language (PSL)

The process specification language process (PSL) is funded by the National Institute of Standards and Technology (NIST). The aim of this project is the development of a common exchange format for production enterprises, that is independent of existing applications, robust enough to represent the necessary process information for any kind of application. The ultimate goal is the support of communication between different applications based on a common understanding of their environment. For process data PSL is designed to become an exchange format like STEP in the domain of product data [27].

The core concept of PSL for the mapping between two application programs is first the mapping of each application's modeling ontology to the PSL ontology. Following this, the source application process can be represented using the Knowledge Interchange Format and transformed into a process that conforms to the PSL ontology. From this intermediate process a target process using the target application's ontology can be created using KIF, which in turn can be imported into the target application.

The basic components of PSL are:

- Activites. These can be generic activities, for example deterministic or nondeterministic procedures, as well as ordering functions over activities, such as creates- or precedes-relationships.
- Objects. These can be either resources, such as human resources or machines, or states, such as pre- or post-activity states.
- Timepoints. These can be used to describe the temporal relationships between activities or the durations of procedures.

Similar to the PIF framework, all concepts of PSL are specified using the Knowledge Interchange Format. Ultimately, the PSL and PIF projects will merge, though a consolidated documentation is not available yet. Besides a prototype implementation the research around PSL currently evaluates mapping possibilities between PSL and XML to broaden the applicability of the language.

2.4 Generalized Process Structure Grammars (GPSG)

While WPDL, PIF and PSL represent IPO-based process modeling languages, Generalized Process Structure Grammars (GPSG) as proposed by GLANCE et al. [13] represent a constraint-based approach to process modeling. For the modeling of a process a specific grammar is constructed, that contains the legal elements of the process as well as their relationships. The grammar spans a process space that contains only the vital constraints and construction rules, everything that need not be restricted by default is left variable for the time of process enactment. A process instance in GPSG is a legal phrase that is constructed using the grammar of the underlying process model. Each GPSG can contain two kinds of rules:

- Activity-centric rules, that separate a process goal into subgoals and attach execution constraints. This process is comparable to the activity analysis and separation known from organizational theory.
- Document-centric rules, which describe the data objects treated in the process.

The application of GPSG for process modeling can be illustrated by a simple example (from [13]). While the sequence of two activities in an IPO-based PDL is mostly denoted as

```
B.start := A.end
```

a GPSG-based process modeling language would contain the following constraints:

```
B.start = A.end
B.end < deadline
B.start = B.end – B.average_duration
```

GPSG-based process definitions thus allow for more flexibility during process enactment, because the processes are not executed following a strict set of control flow paths and conditions, but rather emerge within the process space opened by the process-specific grammar.

A prototype system called FREEFLOW [28] was developed using the GPSG approach for process representation. The complexity of the constraint-based workflow model restrict the modeling of processes to a textual specification of the constraints, while a graphical representation seems difficult due to the many possible paths of the process at runtime. A suitable extension of GPSG for workflow process modeling could be the provision of grammar templates for certain types of workflow operations, such as branching, resource assignment, application invocation and so forth. While promising for (partially) unstructured and runtime-defined ad-hoc processes, that are only partially understood at buildtime, the specification of transactional workflows using GPSG will result in rather complex grammars and might be done more efficiently using IPO-based modeling languages.

2.5 Unified Modeling Language

As opposed to the previous four approaches, that focus on a textual description of processes, the Unified Modeling Language (UML) defines different diagram types for the design object-oriented software systems (cf. in the following [29, 30, 31]. The UML is designed around a small number of core concepts that partially existed before UML was standardized in 1997 (the development of UML had started in 1994 with the unification of the different approaches of Booch's method, the Object Modeling Technique and the Object-Oriented Software Engineering method). UML was targeted at ending the method wars of the 1980s and 1990s, where different developers found existing object-oriented modeling methods only partially satisfactory. The concepts of UML can be extended or specialized by users thus enabling the domain-independent definition of complex systems. The different diagram types of UML allow for the design of different views of a system.

During the past years the interest in object-orientation as a means of providing extendable, scalable and distributable workflow management systems has increased significantly [32] and a number of prototypes were developed (cf. e. g. WASA2 [33] and WORCOS [34]). Starting with the internal architecture of these systems, the necessity for an object-oriented modeling method of workflow models led to the evaluation of UML as a modeling language [35]. As opposed to the other modeling languages presented in this paper, the UML offers graphical notations for workflow models. However, there is not one single notation that can be used to model all aspects of a workflow model. Instead, several diagram types have to be employed in order to model all aspects of a workflow process. These diagram types are in particular:

- *Use Case Diagrams*. These diagrams are used to depict the interaction of a system with its environment. In the case of a workflow process use cases can be used to model the interaction of a process or activity with outside actors (workflow participants, customers, external systems etc.). Use cases denote only the static relationship between actors and system functionality, but do not describe the temporal or logical sequence of process steps. The decomposition of a workflow process into several subprocesses or elementary activities can be modeled using the "uses" relationship between use cases (e. g. an `order management` use case "uses" the use cases `enter customer data, place order` and `settle payment`).
- *Sequence Diagrams*. These diagrams depict the temporal and logical order of activities and involved participants in a "swim lane"-style notation. If the different actors within a workflow process are arranged in parallel lanes, the interaction between participants allow for a mapping to speech-act based workflow models as well as IPO-style workflow models.
- *Collaboration Diagrams*. Within collaboration diagrams the interaction between actors and use cases are described in terms of the messages that are sent between the different elements

of the diagram. Collaboration diagrams can be seen as an extension of the use case diagrams because they allow for an ordering of messages as well as for directed relationships.

- *Statechart Diagrams*. A statechart diagram shows all possible states of a use case and the transitions between these states. Used in the context of workflow management a statechart can be used to depict the possible starting and ending points of a workflow model as well as the legal transitions between states (e. g. ready, started, suspended, finished, terminated etc.). Statecharts can also be used to describe the transformation of process objects in a workflow process, e. g. an invoice in an invoive auditing process.
- *Activity Diagrams.* Activity diagrams are variations of statecharts that display all possible paths of action between activities. While Statecharts may contain passive states, activity diagrams depict relationships between activities. The transition between two activities is only active if the preceeding activity has finished and an optional guard constraint at the transition evaluates to true. Modeling elements allow for parallel branches as well as alternatives between activities. While the original UML notation leads to rather complex diagrams for alternative paths or conditional branching, the next version of the UML will allow for a shorthand notation of these cases.

The current version 1.1 of the UML has some shortcomings with regard to workflow process modeling, that will be partially addressed by the forthcoming version 1.3 which is to be published in late 1999. Some of the deficiencies are (for a detailed discussion cf. [36]):

- Insufficient differentiation between data- and control-flow in an activity diagram
- No operators for event handling in activity diagrams
- Lacking facilities for the modeling of resources (people, roles, organizational units) and their relation to the workflow activities

In total, the UML offers a variety of diagram types that can accommodate several aspects of workflow process modeling. The necessity to use different diagram types that are not always orthogonal and the lack of resource modeling, however, make the current UML standard difficult to use in current workflow tools. With the development of new versions of the UML this situation is likely improve over time.

3 Comparison of the Different Approaches

The different approaches presented in section two are summarized in table 1. This summary contains the origin and target domain of the different approaches, as well as their capability to support different elements of workflow process models.

	WPDL	PIF	PSL	GPSG	UML
Origin	Vendor-driven standardization organization	Academic Project	Governmental standardization organization	Commercial research laboratory	Commercial/ Standardization organization
Target Domain	Workflow processes	Business processes	Manufacturing processes	Business and workflow processes	Object-oriented system development
Specification	EBNF	KIF	KIF	EBNF	OCL, UML, Free Text
Supported Platforms	Div. WF and Modeling Tools	Prototype	Prototype	Prototype	Div. Modeling and CASE Tools
Process Modeling	Textual Notation	Textual Notation	Textual Notation	Extension of a constraint repository with user-defined constraints	Use Cases, Interaction Diagrams, Sequence Diagrams, Activity Diagrams
Resource Modeling	Minimum meta model, interface to external resource management system	Agents as specializations of generic objects, user defined specializations	Several resource concepts as specializations of generic objects	-	Not specifically, only roles
Data Modeling	Reference to external data repositories, elementary data types	Elementary data types (Number, String, Symbol, List, PIF-Sentence)	-	Using document-centric rules	Static Structure Diagrams
Modeling of invoked applications	Reference to external applications	Specializations of objects	Specialization of generic objects	-	Methods of objects invoked in Sequence Diagrams
Extendable	Yes, through user-defined attributes	Yes, through inheritance and user-defined attributes	Yes, through inheritance and user-defined attributes	Yes, through new user-defined constraints	Yes, through user extensions
Target	Interchange Language	Interchange Language	Interchange Language	Modeling Language	Modeling Language

Table 1. Summary of the findings.

4 Summary and Conclusions

We have presented five different meta-languages for the modeling of processes and evaluated their capability to model workflow processes. Each of the languages has a different origin and a different target domain, thus offering specific strengths and weaknesses. While the WPDL was specifically designed for the modeling of workflow processes, it is only extendable with regard to its attributes, thus limiting its application to IPO-based (and to a limited extent speech-act-based) workflow modeling paradigms. In turn, the PIF and PSL approaches are defined around a small set of core components with the inherent capability of being extended for specific user requirements. While the process modeling constructs of PIF are simultaneously simple yet expressive, PSL has strengths with regard to the differentiated representation of resource concepts. The proposed merger of PIF and PSL – if done right – might lead to a very powerful language for process model interchange. In order to make this language applicable to workflow process models, a specific subset should be defined. The GPSG approach is a novel and still little used way of representing processes by opening a multitude of legal process paths within a space that is limited by constraints. A mapping of these concepts to traditional interchange languages or a set of workflow relevant templates are needed to further evaluate the possibilities of this approach. The Unified Modeling Language differs from the previous approaches because it is targeted to a specific programming paradigm (object-orientation), it is a visual language and it consists of different representations for different aspects of process models. The advantage of UML is the existing tool support and the backing of a large number of vendors, namely through the Object Management Group. On the negative side the use of different diagram types for non-orthogonal aspects of a process model leads to redundancies in modeling and still lacks several important aspects for workflow process modeling, such as resource specification or unconstraint parallelism.

With the advance of interorganizational workflow processes and the increasing use of modeling tools and workflow management systems in enterprises, interchange languages such as the presented meta-languages play a significant role in the development of integrated information systems. The existing representations point in the right direction, but further research is needed to eliminate the discussed deficiencies and to improve the applicability of these approaches.

References

[1] Ellis, C. A.; Nutt, G.: Office Information Systems and Computer Science. ACM Computing Surveys 12 (1980) 1, ACM Publishers, New York, 1980, pp. 27-60.

[2] Grudin, J.: CSCW: History and Focus. IEEE Computers 27 (1994) 5, pp. 19-26.

[3] Alonso, G.; Agrawal, D.; El Abbadi, A.; Mohan, C.: Functionalities and Limitations of Current Workflow Management Systems. IEEE Expert Systems 12 (1997) 5.

[4] Georgakopoulos, D.; Hornick, M. F.; Manola, F.; Brodie, M. L.; Heiler, S.; Nayeri, F.; Hurwitz, B.: An Extended Transaction Environment for Workflow in Distributed Object Computing. IEEE Bulletin of the Technical Committee on Data Engineering, 16 (1993) 2.

[5] Sheth, A.; Rusinkiewicz, M.: On Transactional Workflows. IEEE Data Engineering 16 (1993) 2.

[6] Sheth, A.: From contemporary workflow automation to adaptive and dynamic work activity coordination and collaboration. SIGGROUP Bulletin 18 (1997) 3, pp. 17-20.

[7] Jablonski, St.; Bussler, Chr.: Workflow Management Systems – Modeling Concepts, Architecture and Implementations. Thomson International Publishers, London 1996.

[8] Weske, M.: Flexible Modeling and Execution of Workflow Activities. Technical Report Angewandte Mathematik und Informatik 08/97-I, University of Muenster, 1997.

[9] Carlsen, S.: Conceptual Modeling and Composition of Flexible Workflow Models. PhD Thesis, Information Systems Group, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim 1997.

[10] Emmerich, W.; Gruhn, V.: Software Process Modelling with FUNSOFT Nets.

[11] Leymann, F.; Altenhuber, W.: Managing Business Processes as an Information Resource. IBM Systems Journal, 33 (1994) 2, pp. 326-348.

[12] Medina-Mora, R.; Winograd, T.; Flores, R.; Flores, F.: The Action Workflow Approach to Workflow Management Technology. In: CSCW’92: Proceedings of the Conference on Computer Supported Cooperative Work, ACM Publishers, New York, pp. 281-288.

[13] Glance, N. S.; Pagani, D. S.; Pareschi, R.: Generalized Process Structure Grammars (GPSG) for flexible representations of work. In: M. Ackerman (Ed.): CSCW'96: Proceedings of the Conference on Computer Supported Cooperative Work. Boston, MA, 1996, pp. 180-189.

[14] Senge, P.: The fifth discipline – The art and practice of the learning organization. Century Business Publishers, London, 1990.

[15] Batini, C.; Lenzerini, M.; Navathe, S. B.: A Comparative Analysis of Methodologies for Database Schema Integration, ACM Computing Surveys, 18 (1986) 4, pp. 323–364.

[16] Knutilla, A.; Schlenoff, C.; Ray, S.; Polyak, S. T.; Tate, A.; Cheah S. C.; Anderson, R. C.: Process Specification Language: An Analysis of Existing Representations, NISTIR 6160, National Institute of Standards and Technology, Gaithersburg, MD, 1998.

[17] Heinrich, L. J.; Damschik, I.; Gappmaier, M.; Häntschel, I.: Workflow-Management. Produktevaluierung im Labor. In: HMD, 32 (1995) 181, S. 101-112.

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

[19] zur Mühlen, M.: Evaluation of Workflow Management Systems Using Meta Models. In: Sprague, R. Jr. (Ed.): Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences (HICSS 1999). Wailea, January 1999.

[20] Workflow Management Coalition: Terminology and Glossary. Version 3.0. April 1999. Winchester 1999.

[21] Object Management Group: Workflow Management Facility. Document bom 99-03-01. Framingham, MA 1999.

[22] Workflow Management Coalition. Workflow Client Application (Interface 2) Application Programming Interface (WAPI) Specification. Document Number WfMC TC-1009. Winchester 1997.

[23] Workflow Management Coalition. Interface 1: Process Definition Interchange Process Model. Document Number WfMC TC-1016-M. Winchester 1998.

[24] Malone, T.; Crowston, K.; Lee, J.; Pentland, B.: Tools for Inventing Organizations: Toward A Handbook of Organizational Processes. In: 1993 Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative, IEEE Computer Society Press, 1993.

[25] Lee, J.; Gruninger, M.; Jin, Y.; Malone, T.; Tate, A.; Yost, G. et. al.: The PIF Process Interchange Format and Framework Version 1.2. The Knowledge Engineering Review, 13 (1998) 1, Cambridge University Press, March 1998, pp. 91-120.

[26] Genesereth, M. R.: Knowledge Interchange Format - draft proposed American National Standard (dpANS). NCITS.T2/98-004. Online Paper as of 1999-09-04. http://logic.stanford.edu/kif/dpans.html

[27] International Standardization Organization: Product data representation and exchange: Part 1: Overview and fundamental principles, ISO Standard 10303-1, 1992.

[28] Dourish, P.; Holmes, J.; MacLean, A.; Marqvardsen, P.; Zbyslaw, A.: Freeflow: Mediating Between Representation and Action in Workflow Systems. In: M. Ackerman (Ed.): CSCW’96: Proceedings of the Conference on Computer Supported Cooperative Work. Boston, MA, 1996, pp. 190-198.

[29] Rational, Inc. et al.: UML Notation Guide version 1.1. OMG Document ad/97-08-05. Framingham, MA 1997.

[30] Rational, Inc. et al.: UML Semantics version 1.1. OMG Document ad/97-08-04. Framingham, MA 1997.

[31] Rational, Inc. et al.: UML Summary version 1.1. OMG Document ad/97-08-03. Framingham, MA 1997.

[32] Leymann, F.: Workflow make Objects really useful. In: Proceedings of the 6th International Workshop on High Performance Transaction Systems (HPTS). Asilomar, CA 1996.

[33] Weske, M.: Design and Implementation of an Object-Oriented Workflow Management System based on CORBA. Working Paper Nr. 33/98-I of the Institute for Applied Mathematics and Computer Science, University of Münster, Münster 1998.

[34] Schulze, W.; Böhm, M.; Meyer-Wegener, K.: Services of Workflow Objects and Workflow Meta-Objects in OMG-compliant Environments. In: Proceedings of the OOPSLA’96 Workshop on Business Object Design and Implementation, San Jose, CA 1996.

[35] Hruby, P.: Specification of Workflow Management Systems with UML. Proceedings of the 1998 OOPSLA Workshop on Implementation and Application of Object-oriented Workflow Management Systems, Vancouver, BC 1998.

[36] Wiegert, O.: Business Process Modeling and Workflow Definition with UML – Deficiencies and Actions to Improve. Presentation at the OMG Meeting Manchester, 1998-03-31.