Perspectives to Process Modeling – A Historical Overview

John Krogstie

Norwegian University of Science and Technology
krogstie@idi.ntnu.no

Abstract. Processes modeling is done for a number of reasons in relation to enterprise modeling, business process modeling and information systems development in general, and this paper will give an overview of main approaches to different types of process modeling. Modeling approaches are structured according to the main modeling perspective being used. In conceptual modeling in general, one can identify 8 modeling perspectives; behavioral, functional, structural, goal-oriented, object-oriented, language action, organizational and topological. In the paper we will present both historical and current examples of process modeling according to these different perspectives, and discuss what perspectives are most appropriate to achieve the different goals of modeling.

Keywords: Process modeling, conceptual modeling.

1 Introduction

A *process* is a collection of related, structured tasks that produce a specific service or product to address a certain goal for a particular actor or set of actors. Process modeling has been performed relative to IT and organizational development at least since the 70ties. The interest has gone through phases with the introduction of different approaches, including Structured Analysis in the seventies [33], BPR in the late eighties/early nineties [42], and Workflow Management in the nineties [95]. Lately, with the proliferation of BPM (Business process management) [46], interest and use of process modeling has increased even further, although focusing primarily on a selected number of modeling approaches.

Models of work processes have long been utilized to learn about, guide and support practice also in a number of areas. In software process improvement [22], enterprise modeling [32] and quality management, process models describe methods and standard working procedures. Simulation and quantitative analyses are performed to improve efficiency [7, 61]. In process centric software engineering environments [9] and workflow systems [95] model execution is automated. This wide range of applications is reflected in current modeling languages, which emphasize different aspects of the process.

The archetypical way to look on processes is as a transformation, according to an IPO (input-process-output) approach. Whereas early process modeling approaches had this as a basic approach [33], as process modeling have been integrated with other types of conceptual modeling, variants of this have appeared.

I. Bider et al. (Eds.): BPMDS 2012 and EMMSAD 2012, LNBIP 113, pp. 315–330, 2012.

First we describe different reasons for doing process modeling. Then we describe different perspectives to modeling, before we in section 4 provide a brief overview of modeling languages used for process modeling following the different perspectives. Since many of those languages being used in practice are developed a long time ago [20] or are extensions of these, we provide also a partly historical overview. In the conclusion we briefly summarize how modeling according to the different perspectives is beneficial to achieve the various goals of modeling. Since the different goals of modeling require different properties from the modeling language used, it is useful to look more closely on the properties of different modeling perspectives to be able to choose an appropriate modeling approach. Due to size limitation of this paper, this overview will only be on a high level.

2 Application of Process Modeling

According to general model theory [87] there are three common characteristics of models: *Representation, Simplification* and *Pragmatic orientation.*

- *Representation*: Models are models of something else
- *Simplification*: Models possess a reductive trait in that they map a subset of attributes of the phenomenon being modeled
- *Pragmatic orientation*: Models have a substitutive function in that they substitute a certain phenomenon as being conceptualized by a certain subject in a given temporal space with a certain incentive or operation in mind

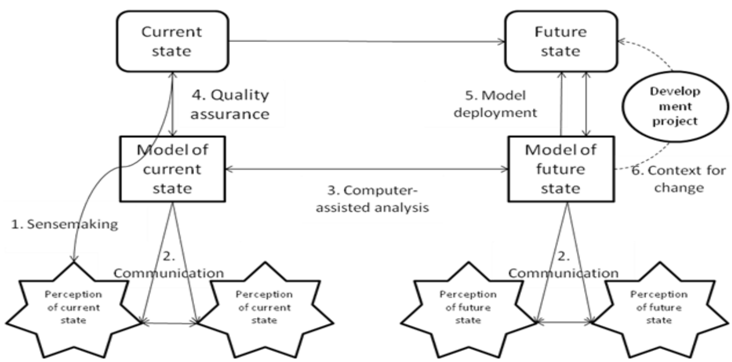

Fig. 1. Organizational application of modeling

Process modeling is usually done in some organizational setting. As illustrated in Fig. 1 one can look upon an organization and its information system abstractly to be in a state (the current state, often represented as a descriptive 'as-is' model) that are to be evolved to some future wanted state (often represented as a prescriptive 'to be' model). Obviously, changes will happen in an organization no matter what is actually planned, thus one might in practice have the use for many different models and scenarios of possible future states, but we simplify the number of possible future states in the discussion below.

The state includes the existing processes, organization and computer systems. These states are often modeled, and the state of the organization is perceived (differently) by different persons through these models. Different usage areas of conceptual models as described in [60, 73]:

1. Human sense-making: The descriptive model of the current state can be useful for people to make sense of and learn about the current perceived situation.
2. Communication between people in the organization: Models can have an important role in human communication. Thus, in addition to support the sense-making process for the individual, a model can act as a common framework supporting communication both relative to descriptive and prescriptive models.
3. Computer-assisted analysis: This is used to gain knowledge about the organization through simulation [6] or deduction, often by comparing a model of the current state and a model of a future, potentially better state.
4. Quality assurance, ensuring e.g. that the organization acts according to a certified process developed for instance as part of an ISO-certification process.
5. Model deployment and activation: To integrate the model of the future state in an information system directly, making the prescriptive model the descriptive model. Models can be activated in three ways:
   a. Through people, where the system offers no active support.
   b. Automatically, where the system plays an active role, as in most automated workflow systems.
   c. Interactively, where the computer and the users co-operate [56].
6. To be a prescriptive model to be used in a traditional system development project, without being directly activated.

3 Perspectives to Modeling

Modeling languages can be divided into classes according to the core phenomena classes (concepts) that are represented and focused on in the language. This has been called the *perspective* of the language [60, 62]. Languages in different perspectives might overlap in what they express, but emphasize different concepts as described below. A classic distinction regarding modeling perspectives is between the structural, functional, and behavioral perspective [74]. Object-orientation analysis appeared as a particular way of combining the structural and behavioral perspective in the late eighties.

Through other work, such as [19], [70], F3 [15], NATURE [51], [57] additional perspectives have been identified, including goal, actor, communicational, and topological. To provide a broad overview of the different perspectives conceptual modeling approaches accommodate, we look on the following:

Behavioral Perspective: Languages following this perspective go back to the early sixties, with the introduction of Petri-nets [79]. In most languages with a behavioral perspective the main phenomena are 'states' and 'transitions' between 'states'. State transitions are triggered by 'events' [21].

Functional Perspective: The main phenomena class in the functional perspective is the 'transformation': A transformation is defined as an activity which based on a set of phenomena transforms them to another set of phenomena.

Structural Perspective: Approaches within the structural perspective concentrate on describing the static structure of a system. The main construct of such languages is the 'entity'.

Goal and Rule Perspective: Goal-oriented modeling focuses on 'goals' and 'rules'. A rule is something which influences the actions of a set of actors. A rule is either a rule of necessity or a deontic rule [58, 75]. A rule of necessity is a rule that must always be satisfied. A deontic rule is a rule which is only socially agreed among a set of persons and organizations. In the early nineties, one started to model so-called rule hierarchies, linking rules of different abstraction levels.

Object-Oriented Perspective: The basic phenomena of object oriented modeling languages are similar to those found in most object oriented programming languages; 'Objects' with unique id and a local state that can only be manipulated by calling methods of the object. Objects have a life cycle. The process of the object is the trace of the events during the existence of the object. A set of objects that share the same definitions of attributes and operations compose an object class.

Communication Perspective: The work within this perspective is based on language/action theory from philosophical linguistics. The basic assumption of language/action theory is that persons cooperate within work processes through their conversations and through mutual commitments taken within them.

Actor and Role Perspective: The main phenomena of languages within this perspective are 'actor' (also termed agent) and 'role'. The background for modeling in this perspective comes both from organizational science, work on programming languages, and work on intelligent agents in artificial intelligence.

Topological Perspective: This perspective relates to the topological ordering between the different concepts. The best background for conceptualization of these aspects comes from the cartography and CSCW fields, differentiating between space and place [28, 45]. 'Space' describes geometrical arrangements that might structure, constrain, and enable certain forms of movement and interaction; 'place' denotes the ways in which settings acquire persistent social meaning through interaction.

4 Perspectives to Process Modeling

We here provide a very brief overview of process modeling according to the different modeling perspectives identified in section 3 above.

4.1 Process Modeling According to the Behavioral Perspective

States (of systems, products, entities, processes) and transformations between states are the central concepts in this perspective. There are two language-types commonly used to model states: State transition diagrams (STD) and state transition matrices (STM). The vocabulary of state transition diagrams is

- State: A system is always in one of the states in the lawful state space for the system. A state is defined by the set of transitions leading to that state, the set of transitions leading out of that state and the set of values assigned to attributes of the system while the system resides in that state.
- Event: An event is a message from the environment or from system itself to the system. The system can react to a set of predefined events.
- Condition: A condition for reacting to an event.
- Transition: Receiving an event will cause a transition to a new state if the event is defined for the current state, and if the condition assigned to the event evaluates to true.
- Action: The system can perform an action in response to an event.

It is generally acknowledged that a large complex system cannot be described in a flat state-model, because of the unmanageable, exponentially growth of states. Hierarchical abstraction mechanisms were added to traditional STDs in Statecharts [43]. Statecharts are integrated with functional modeling (see below) in [44]. Later extensions of Statecharts for object-oriented modeling were developed through the nineties, and Statecharts are the basis for the state transitions diagrams in UML (for the modeling of object-states) [14].

Petri-nets [79] are another well-known behavior-oriented modeling language. Here, *places* indicate a system state space, and a combination of *tokens* located in the places determines the specific system state. State transitions are regulated by firing rules: A transition is enabled if each of its input places contains a token. A transition can fire at any time after it is enabled. The transition takes zero time. After the firing of a transition, a token is removed from each of its input places and a token is produced in all output places. Control-flow aspects like precedence, concurrency, synchronization, exclusiveness, and iteration can be modeled in a Petri-net. There exists several dialects of the Petri net language (going back to [67]) where the transitions are allowed to take time, and these approaches provide decomposition in a way not very different from that of a data flow diagram. Timed Petri Nets [67] also provide probability distributions that can be assigned to the time consumption of each transition and are particularly suited to performance modeling. Other variants are tokens with named and typed variables (Colored Petri Nets), and nets where transitions have pre- and post-conditions in some logic. Colored Petri nets are used in particular for simulation and analysis [52].

Another type of behavioral modeling is based on System dynamics. Systems thinking [85] regards causal relations as mutual, circular and non-linear, hence the straightforward sequences in transformational process models is seen as an idealization that hides important facts. This perspective is also reflected in mathematical models of interaction [93]. System dynamics have been utilized for analysis of complex relationships in cooperative work arrangements [7]. System dynamic process models can be used for analysis and simulation, but not for model activation. A challenge is that it can be difficult to find data to run simulations.

4.2 Process Modeling According to the Functional Perspective

Most popular process modeling languages take a functional (or transformational / input-process-output) approach [20]; although some of the most popular recent languages also include behavioral aspects as will be discussed below. Processes are often divided into activities, which may be divided further into sub-activities. Each activity takes inputs, which it transforms to outputs. Input and output relations thus define the sequence of work. This perspective is chosen for the standards of the Workflow Management Coalition [95], the Internet Engineering Task Force (IETF) [13] as well as most commercial systems [30]. IDEF-3x [50] and Data Flow Diagram (DFD) [33] are paradigm examples of this. DFDs describe a situation using: Processes, data stores, flows, and external entities.

When a process is decomposed into a set of sub-processes, the sub-processes are co-operating to fulfill the higher-level function. This view on DFDs has resulted in the “context diagram" that regards the whole system as a process which receives and sends all inputs and outputs to and from the system. A context diagram determines the boundary of a system. A variant of context-diagrams is Use Case diagrams [14].

DFD and use-cases are semi-formal languages. Some of the short-comings of DFD regarding formality were first addressed in the transformation schema presented by Ward [92] including both data and control transformations, data and event flows (signals, activation and deactivation) (data flows being either discrete or continuous) and variants of stores. A number of the recent process modeling notations typically add control-flow aspects to a transformational approach and combine aspects of the functional and behavioral perspectives. Some examples of this are ARIS EPC, UML Activity Diagrams, YAWL [90], and BPMN.

An Event-driven Process Chain (EPC) [54] is a graphical modeling language used for business process modeling. EPC was developed within the framework of Architecture of Integrated Information System (ARIS) [81] to model business processes. The strength of EPC lies on its simple notation that is capable of portraying business information system while at the same time incorporating other important features such as functions, data, organizational structure, and information resources. However the semantics of an event-driven process chain are not well defined and it is not possible to check the model for consistency and completeness. As demonstrated in [3], these problems can be partly addressed by translating EPC-models to Petri nets since Petri nets have formal semantics enabling analysis techniques.

The UML Activity diagram is one of the three diagram types in the UML for modeling behavior aspect of systems [14]. The most important concepts in the UML activity diagram are activities, decision, start (split) or end (join) of concurrent activities, and start and end states

In 2004, BPMN was presented as the standard business process modeling notation [96]. Since then BPMN has been evaluated in different ways by the academic community [1, 80] and has become widely supported in industry.

The Business Process Modeling Notation (BPMN version 1.0) was adopted by OMG for ratification in February 2006. The BPMN 2.0 specification was formally released January 2011 [76].

Given the extensive use of functional languages, a number of analyses focus on this category [18, 19, 40]. The expressiveness of these languages typically includes decomposition, and data flow, while organizational modeling and roles often are integrated and given less emphasis. In approaches which integrate behavioral and functional aspects, we see also a support for control flow. Aspects like timing and quantification, products and communication, or commitments are better supported by other perspectives. User-orientation is a major advantage of transformational languages, in particular the pure functional ones. Graphical input-process-output models are comprehensible given some training, but you can also build models by simply listing the tasks in plain text, or in a hierarchical work breakdown structure.

4.3 Process Modeling According to the Structural Perspective

The structural perspective has traditionally been handled by languages for data modeling, but also includes approaches from semantic networks and the semantic web. In ER-modeling as described by [17], the basic components are:

- Entities. An entity is a phenomenon that can be distinctly identified. Entities can be classified into entity classes
- Relationships. A relationship is an association among entities. Relationships can be classified into relationship classes
- Attributes and data values. To give value to a property of an entity or relationship. Values are grouped into value classes by their types.

Structural modeling is often perceived to be fundamentally different from functional (process) modeling, since it focus on the static aspects, whereas process modeling focus on dynamics. It is possible to look at processes as entities though (like one have done in object-oriented process modeling discussed below, looking at the process instances as the objects) it which case one can model the situation in a similar way as when doing more traditional data-modeling.

One finds very few attempts on pure structural process modeling in practice, although as we will discuss below, there are approaches to object-oriented process modeling.

4.4 Process Modeling According to the Goal and Rule Perspective

In the workflow area, the use of rules for guiding the workflow is often termed declarative workflow. Constraint based languages [27, 35] prescribe a course of events, rather they capture the boundaries within which the process must be performed, leaving the actors to control the internal details. Instead of telling people what to do, these systems warn about rule violations and enforce constraints. Thus, problems with over-serialization can be avoided [35].

A wide variety of declarative modeling approaches has been specified in business process management, from the use of basic Event-condition-action (ECA)-rules [53] to declarative process modeling languages such as DecSerFlow [4], BPCN [66] and ConDec [78]. In [36] an overview of the most common declarative process modeling languages can be found.

Several advantages have been experienced with a declarative, rule-based approach to information systems modeling [59], but also a number of challenges. Languages representing rule-based process modeling can potentially provide a higher expressiveness than diagrammatic languages (e.g. the ability to specify temporal requirements) [66], but this might result in process models which are less comprehensible [29].

Declarative process enactment guarantees high run-time flexibility for declarative process specifications that contain only the strictly required mandatory constraints. An individual execution path that satisfies the set of mandatory constraints can be dynamically built for a specific process instance. Process compliance is assured when all mandatory rules are correctly mapped onto mandatory business constraints. During the construction of a suitable execution path little support is provided to the end user [94], which could affect the process effectiveness. In [58] differentiating constraints by modality is proposed, recommendations were introduced to guide the user whereas obligations would ensure compliant behavior. Lastly, the increased size and complexity of contemporary process models might decrease the potential for process automation since current declarative workflow management systems might have limited efficiency in when having to take into account a large number of rules according to [5].

A graphic depiction is difficult since it would correspond to a visualization of several possible solutions to the set of constraint equations constituting the model. The support for articulation of planned and ongoing tasks is limited. Consequently, constraints are often combined with transformational models [27, 55, 63]. Alternatively one can have the operational rules related to the process model also linked to goal hierarchies as in [58, 59].

4.5 Process Modeling According to the Object-Oriented Perspective

UML [14] has become both the official and de facto standard for object oriented analysis and design. Consequently, people also apply UML to model business processes. Object orientation offers a number of useful modeling mechanisms like encapsulation, polymorphism, subtyping and inheritance [64, 71]. UML integrates these capabilities with e.g. requirements capture in use case descriptions as described above and behavior modeling in state, activity and sequence diagrams. On the other hand, UML is designed for software developers, not for end users. A core challenge thus remains in mapping system-oriented UML constructs to user- and process-oriented concepts [47]. To this problem no general solution exists [64]. One approach which is somewhat similar to how one can use structural modeling for process modeling is PML [10]. Here one uses object oriented techniques based on looking upon classes in a particular way. Whereas a class is defined by <class name, attributes, methods>, in PML one define this as <process name, methods, resources>. The PML process class describes the process in a generic way. It allows one to define all methods with assurances and resources needed for the process. The instantiation of a process is a project. This means, the instance of a process defines the current occurrence of resources, used data models etc. Regarding connections and dependencies between single process classes, PML features the standard UML-mechanisms of inheritance and associations.

Although with intriguing possibilities, it is safe to say that full-fledged OO process modeling has yet to be taken into use in large scale in practice.

4.6 Process Modeling According to the Communication Perspective

The communication perspective, often termed the language action perspective was brought into the workflow arena through the COORDINATOR prototype [97], later succeeded by the Action Workflow system [69]. This perspective is informed by speech act theory [82], which extends the notion that people use language to describe the world with a focus on how people use language for coordinating action and negotiating commitments. Habermas took Searle's theory as a starting point for his theory of communicative action [41]. Central to Habermas is the distinction between strategic and communicative action. When involved in strategic action, the participants try to achieve their own private goals. When they cooperate, they are only motivated empirically to do so. When involved in communicative action, the participants are oriented towards mutual agreement. The motivation for co-operation is thus rational. Illocutionary logic [26, 84] is a logical formalization of the theory of Searle. The main parts of illocutionary logic are the illocutionary act consisting of three parts, illocutionary context, illocutionary force, and propositional context. The context of an illocutionary act consists of five elements: Speaker, hearer, time, location, and circumstances. The illocutionary force determines the reasons and the goal of the communication. The central element of the illocutionary force is the illocutionary point, and the other elements depend on this. Five illocutionary points are distinguished [83]: Assertives, Directives, Commissives, Declaratives, Expressives

Speech act theory is the basis for modeling of workflow as coordination among people in Action Workflow [69]. The main strength of this approach is that it facilitates analysis of the communicative aspects of the process. It highlights that each process is an interaction between a customer and a performer, represented as a cycle with four phases: preparation, negotiation, performance and acceptance. The dual role constellation is a basis for work breakdown, e.g. the performer can delegate parts of the work to other people. This explicit representation of communication and negotiation, and especially the structuring of the conversation into predefined speech act steps, has also been criticized [16, 23, 88]. Minimal support for situated conversations, the danger that explication leads to increased external control of the work, and a simplistic one-to-one mapping between utterances and actions are among the weaknesses pointed to. On the other hand, it has been reported that the Action Workflow approach is useful when people act pragmatically and don't always follow the encoded rules of behavior [23], i.e. when the communication models are interactively activated.

Some approaches to workflow modeling combine aspects of both the functional and communicative perspective. In WooRKS [8] functional modeling is used for processes and language action modeling for exceptions. TeamWare Flow [89] on the other hand can be said to be a hybrid approach. In addition to the approach to workflow-modeling described above, several other approaches to conceptual modeling are inspired by the theories of Habermas and Searle such as SAMPO [11], and ABC/DEMO [24, 25].

4.7 Process Modeling According to the Actor and Role Perspective

Role-centric process modeling languages have been applied for work-flow analysis and implementation. Role Interaction Nets (RIN) [86] and Role Activity Diagrams (RAD) [77] use roles as a main structuring concept. The activities performed by a role are grouped together in the diagram, either in swimlanes (RIN), or inside boxes (RAD). The use of roles as a structuring concept makes it very clear who is responsible for what. RAD has also been merged with speech acts for interaction between roles [12]. A newer approach in this direction is S-BPM (subject-oriented business process management [31]).

The role-based approach also has limitations, e.g. making it difficult to change the organizational distribution of work. It primarily targets analysis of administrative procedures, where formal roles are important. The use of swimlanes in BPMN and UML Activity Diagrams described above might also have this effect. Some other approaches worth discussing here are REA and e^3Value.

The REA language was first described in McCarthy [68]. It has been developed further in [34]. REA was originally intended as a basis for accounting information systems and focuses on representing increases and decreases of value in an organization. REA has later been extended to apply to enterprise architectures [49] and e-commerce frameworks [91].

The core concepts in the REA language are *resource*, *event* and *agent*. The intuition behind this language is that every business transaction can be described as an event where two agents exchange resources. In order to acquire a resource from other agents, an agent has to give up some of its own resource. It seldom happens that one agent simply gives away a resource to another without expecting another resource back as compensation. Basically, there are two types of events: *exchange* and *conversion* [49]. An exchange occurs when an agent receives economic resources from another agent and gives resource back to that agent. A conversion occurs when an agent consumes resources to produce other resources. REA has influence the electronic commerce standard ebXML.

E^3Value [39] is an actor/role oriented modeling language for inter-organizational modeling. The purpose of this modeling language is to represent how actors of a system create, exchange and consume objects of economic value, only including value-adding activities. The modeling language focuses on the key points of a business model, to get an understanding of business operations and systems requirements through scenario analysis and evaluation. The purpose of e^3value is to determine whether a business idea is profitable or not, that is to say by analyzing for each actor involved in the system if the idea is profitable for them or not. E^3value models give a representation of actors, exchanges, value objects of a business system. Modeling at the actor-level is one approach to address BPM-in-the-large [48].

4.8 Process Modeling According to the Topological Perspective

The concept of place can be related to a process, given that a place focuses on the typical behavior in a certain setting rather than where this is physically. Whereas some processes are closely related to place (e.g. what can be done in a certain, specialized factory), more and more tasks can be done in more or less any setting due to the mobile communication infrastructure, thus making it useful to be able to differentiate geographic/topological from transformation-oriented modeling. In certain representations, aspects of space and place is closely interlinked (e.g. in the representation of the agenda of a conference, also taking time into account). Some approaches letting you take the place into account exists, e.g. work on extending UML activity diagrams with place-oriented aspects [3]. An even more topologically oriented approach is to group concepts at the same location [38].

Traditional representations of space such as a map have to a limited degree been oriented towards representation of process knowledge. Some recent approaches do take these aspects more consciously into account, as exemplified by [72], combining conceptual, temporal, and geographic knowledge representation. Other approaches use the topological perspective more as a metaphor [2].

5 Concluding Remarks

We have summarized this high-level overview of the field, looking upon approaches according to different perspectives relative to the different usage areas for process modeling presented in section 2, and also indicated the amount of actual use of the approach in practice.

Table 1. Usage of modeling perspectives

Area (vs. Fig. 1)	1+2	3	4	5a	5b	5c	6
Perspective (vs. 4.1-4.8)	Sense& Com	An al.	QA	Man. Activa-tion	Work -flow	Inter-active activ.	Req. for ISD
4.1 Behavioral	-/o	+/o	-/-	-/-	+/o	-/-	o/-
4.2 Functional	+/+	-/-	+/o	+/o	-/-	o/-	+/o
4.2 Behavioral + Functional	+/o	+/o	o/-	o/-	+/+	+/o	o/o
4.3 Structural	o/-	-/-	-/-	-/-	-/-	-/-	o/-
4.4 Rule/Goal	-/-	o/-	o/-	-/-	o/-	o/-	o/o
4.5 Object-oriented	-/-	o/-	-/-	-/-	-/-	-/-	o/-
4.6 Communicational	o/-	-/-	-/-	-/-	o/-	o/-	-/-
4.7 Actor	+/o	o/-	o/-	o/-	-/-	-/-	o/o
4.8 Topological	o/-	-/-	-/-	o/-	-/-	o/-	o/-

The legend indicates the applicability of the approach / actual use of the approach (relative to the usage of modeling for this task), '+' indicates good applicability or high use, 'o' is some applicability and use, whereas '-' indicate poor applicability and limited use. E.g. o/- under the communicational perspective for sense-making and communication indicates that it has some applicability for this use, but are very little used in practice. Obviously different approaches according to the same perspective can be more or less applicable, and different languages of a certain perspective would score differently based on the concrete expressiveness and level of formality of the language and modeling approach. Due to space limitations, it is not possible to provide the detailed concrete evaluations of all approaches that we mentioned in the previous section. From the table, we see that functional and combinations of functional and behavioral approaches are used the most. All other perspectives have potential for use for certain areas, although this often varies relative to concrete needs in the domain for representing particular aspects (such as topological aspects which in many cases might not be relevant). In particular some of the less traditional approaches appear to have large untapped potential for a richer more appropriate representation of what we term processes and business processes. We work on a longer article including examples to better illustrate the pros and cons of different approaches to process modeling.

References

1. Aagesen, G., Krogstie, J.: Analysis and design of business processes using BPMN. In: Handbook on Business Process Management. Springer (2010)
2. van der Aalst, W.M.P.: TomTom for Business Process Management (TomTom4BPM). In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 2–5. Springer, Heidelberg (2009)
3. van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains. Information and Software Technology 41, 639–650 (1999)
4. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: Towards a truly declarative service flow language. In: Web Services and Formal Methods, pp. 1–23 (2006)
5. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science-Research and Development 23(2) (2009)
6. van der Aalst, W.M.P., Nakatumba, J., Rozinat, A., Russel, N.: Business process simulation. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1. Springer (2010)
7. Abdel-Hamid, T.K., Madnick, S.E.: Lessons Learned from Modeling the Dynamics of Software Development. Communications of the ACM 32(12) (2000)
8. Ader, M., Lu, G., Pons, P., Monguio, J., Lopez, L., De Michelis, G., Grasso, M.A., Vlondakis, G.: Woorks, an object-oriented workflow system for offices. Technical report, ITHACA (1994)
9. Ambriola, V., Conradi, R., Fuggetta, A.: Assessing Process-Centered Software Engineering Environments. ACM TOSEM 6(3) (1997)
10. Anderl, R., Raßler, J.: Computer-Aided Innovation (CAI), Gaetano Cascini. IFIP, vol. 277. Springer, Boston (2008)
11. Auramäki, E., Hirschheim, R., Lyytinen, K.: Modelling offices through discourse analysis: The SAMPO approach. The Computer Journal 35(4), 342–352 (1992)
12. Benson, I., Everhard, S., McKernan, A., Galewsky, B., Partridge, C.: Mathematical Structures for Reasoning about Emergent Organization. In: ACM CSCW Workshop: Beyond Workflow Management, Philadelphia, USA (2000)
13. Bolcer, G., Kaiser, G.: SWAP: Leveraging the web to manage workflow. IEEE Internet Computing 3(1) (1999)
14. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language: User Guide. Addison-Wesley (2005)

15. Bubenko Jr. J. A., Rolland, C., Loucopoulos, P., DeAntonellis, V.: Facilitating fuzzy to formal requirements modeling. In: Proceedings of the First International Conference on Requirements Engineering (ICRE 1994), April 18-22, pp. 154–157. IEEE Computer Society Press, Colorado Springs (1994)
16. Button, G.: What's Wrong with Speech Act Theory. CSCW 3(1) (1995)
17. Chen, P.P.: The entity-relationship model: Towards a unified view of data. ACM Transactions on Database Systems 1(1), 9–36 (1976)
18. Conradi, R., Jaccheri, M.L.: Process Modelling Languages. In: Derniame, J.-C., Kaba, B.A., Wastell, D. (eds.) Promoter-2 1998. LNCS, vol. 1500, pp. 27–52. Springer, Heidelberg (1999)
19. Curtis, B., Kellner, M.I., Over, J.: Process Modeling. CACM 35(9) (1992)
20. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data & Knowledge Engineering 58, 358–380 (2006)
21. Davis, A.M.: A comparison of techniques for the specification of external system behavior. Communications of the ACM 31(9), 1098–1115 (1988)
22. Derniame, J.-C., Kaba, B.A., Wastell, D. (eds.): Promoter-2 1998. LNCS, vol. 1500. Springer, Heidelberg (1999)
23. De Michelis, G., Grasso, M.A.: Situating Conversations within the Language/Action Perspective: The Milan Conversation Model. In: ACM CSCW Conference, Chapel Hill, North Carolina, USA (1994)
24. Dietz, J.L.G.: Integrating management of human and computer resources in task processing organizations: A conceptual view. In: HICCS'27, Maui, Hawaii, US, January 4-7 (1994)
25. Dietz, J.L.G.: Enterprise Ontology – Theory and Methodology. Springer (2006)
26. Dignum, F., Weigand, H.: Communication and deontic logic. In: Wieringa, R., Feenstra, R. (eds.) Working papers of the International Workshop on Information Systems - Correctness and Reuseability, IS-CORE 1994 (1994)
27. Dourish, P., Holmes, J., MacLean, A., Marqvardsen, P., Zbyslaw, A.: Freeflow: Mediating between representation and action in workflow systems. In: ACM CSCW Conference, Boston, USA (1996)
28. Dourish, P.: Re-Space-ing Place: "Place" and "Space" Ten Years. In: Proc. of ACM Conf. Computer-Supported Cooperative Work CSCW 2006, Banff, Canada, pp. 299–308. ACM (2006)
29. Fickas, S.: Design issues in a rule-based system. Journal of Systems and Software 10(2), 113–123 (1989)
30. Fischer, L.: Excellence in Practice IV - Innovation and excellence in workflow and knowledge management. Workflow Management Coalition, Future Strategies, USA (2000)
31. Fleischmann, A.: What is S-BPM? In: Buchwald, H., Fleischmann, A., Seese, D., Stary, C. (eds.) S-BPM ONE. CCIS, vol. 85, pp. 85–106. Springer, Heidelberg (2010)
32. Fox, M.S., Gruninger, M.: Enterprise modeling. AI Magazine (2000)
33. Gane, C., Sarson, T.: Structured Systems Analysis: Tools and Techniques. Prentice Hall (1979)
34. Geerts, G.L., McCarthy, W.E.: An Accounting Object Infrastructure for Knowledge-Based Enterprise Models. IEEE Intelligent Systems 14, 89–94 (1999)
35. Glance, N.S., Pagani, D.S., Pareschi, R.: Generalized Process Structure Grammars (GPSG) for Flexible Representation of Work. In: ACM CSCW Conference, Boston, USA (1996)
36. Goedertier, S., Vanthienen, J.: An overview of declarative process modeling principles and languages. Com. of Systemics and Informatics World Network 6, 51–58 (2009)

37. Gopalakrishnan, S., Krogstie, J., Sindre, G.: Adapting UML Activity Diagrams for Mobile Work Process Modelling: Experimental Comparison of Two Notation Alternatives. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 145–161. Springer, Heidelberg (2010)
38. Gopalakrishnan, S., Sindre, G.: Diagram Notations for Mobile Work Processes Presented at PoEM, Oslo Norway November 2-3 (2011)
39. Gordijn, J., Yu, E., van der Raadt, B.: e-service Design using i* and e3value, IEEE Software (May- June 2006)
40. Green, P., Rosemann, M.: Integrated Process Modeling: An Ontological Evaluation. Information Systems 25(3) (2000)
41. Habermas, J.: The Theory of Communicative Action. Beacon Press (1984)
42. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Harper Business (1993)
43. Harel, D.S.: A visual formalism for complex systems. Science of Computer Programming (8), 231–274 (1987)
44. Harel, D., Lachover, H., Naamed, A., Pnueli, A., Politi, M., Sherman, R., Shtull-Trauring, A., Trakhtenbrot, M.: STATEMATE: a working environment for thedevelopment of complex reactive systems. IEEE TSE 16(4), 403–414 (1990)
45. Harrison, S., Dourish, P.: Re-Place-ing Space: The Roles of Space and Place in Collaborative Systems. In: Proc. CSCW 1996, Boston, MA, pp. 67–76. ACM, New York (1996)
46. Havey, M.: Essential Business Process Modelling. O'Reilly (2005)
47. Hommes, B.-J., van Reijswoud, V.: The quality of business process modelling techniques. In: Conference on Information Systems Concepts, Leiden, Nertherlands. Kluwer (1999)
48. Houy, C., Fettke, P., Loos, P., van der Aalst, W.M.P., Krogstie, J.: BPM-in-the-Large – Towards a higher level of abstraction in Business Process Management. Paper Presented at GISP under WCC 2010, Brisbane, Australia (September 2010)
49. Hruby, P.: Model-Driven Design Using Business Patterns. Springer, New York (2006)
50. IDEF-3x Process Modeling Language Specification, Standard NA-94-1422B, Rockwell International (1993)
51. Jarke, M., Bubenko Jr., J., Rolland, C., Sutcliffe, A., Vassiliou, Y.: Theories underlying requirements engineering: An overview of NATURE at genesis. In: Proceedings of RE 1993, pp. 19–31 (1993)
52. Jensen, K., Kristiansen, L.M., Wells, L.: Coloured petri nets and CPN tools got modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transfer 9(3-4), 213–254 (2007)
53. Kappel, G., Rausch-Schott, S., Retschitzegger, W.: Coordination in workflow management systems a rule-based approach. In: Coordination Technology for Collaborative Applications, pp. 99-119 (1998)
54. Keller, G., Nüttgens, M., Scheer, A.W.: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK) (1992)
55. Krogstie, J., McBrien, P., Owens, R., Seltveit, A.H.: Information Systems Development Using a Combination of Process and Rule Based Approaches. In: Andersen, R., Sølvberg, A., Bubenko Jr., J.A. (eds.) CAiSE 1991. LNCS, vol. 498, pp. 319–335. Springer, Heidelberg (1991)
56. Krogstie, J., Jørgensen, H.D.: Interactive Models for Supporting Networked Organisations. In: Persson, A., Stirna, J. (eds.) CAiSE 2004. LNCS, vol. 3084, pp. 550–563. Springer, Heidelberg (2004)
57. Krogstie, J., Opdahl, A., Brinkkemper, S. (eds.): Conceptual Modelling in Information Systems Engineering. Springer (2007)

58. Krogstie, J., Sindre, G.: Utilizing deontic operators in information systems specifications. Requirement Engineering Journal 1, 210–237 (1996)
59. Krogstie, J.: Integrated Goal, Data and Process modeling: From TEMPORA to Model-Generated Work-Places. In: Johannesson, P., Søderstrøm, E. (eds.) Information Systems Engineering From Data Analysis to Process Networks, pp. 43–65. IGI Publishing (2008)
60. Krogstie, J.: Model-based Development and Evolution of Information Systems: A Quality Approach. Springer (2012)
61. Kuntz, J.C., Christiansen, T.R., Cohen, G.P., Jin, Y., Levitt, R.E.: The virtual design team: A computational simulation model of project organizations. Communications of the ACM 41(11) (1998)
62. Lillehagen, F., Krogstie, J.: Active Knowledge Modeling of Enterprises. Springer (2008)
63. Lindland, O.I., Krogstie, J.: Validating Conceptual Models by Transformational Prototyping. In: Rolland, C., Cauvet, C., Bodart, F. (eds.) CAiSE 1993. LNCS, vol. 685, pp. 165–183. Springer, Heidelberg (1993)
64. Loos, P., Allweyer, T.: Process orientation and object-orientation - An approach for integrating UML with event-driven process chains (EPC), Germany (1998)
65. Lu, R., Sadiq, S., Governatori, G.: On managing business processes variants. Data & Knowledge Engineering 68(7), 642–664 (2009)
66. Lu, R., Sadiq, W.: A Survey of Comparative Business Process Modeling Approaches. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 82–94. Springer, Heidelberg (2007)
67. Marsan, M.A., et al. (eds.): Proceeding of the International workshop on Timed Petri Nets, Torino, Italy. IEEE Computer Society Press (1985)
68. McCarthy, W.E.: The REA accounting model: a generalized framework for accounting systems in a shared data environment. The Accounting Review 57, 554–578 (1982)
69. Medina-Mora, R., Winograd, T., Flores, R., Flores, F.: The Action Workflow approach to workflow management technology. In: ACM CSCW Conference (1992)
70. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, É., Elabed, L., El Boussaidi, G.: Business process modeling languages: Sorting through the alphabet soup. ACM Comput. Surv. 43(1), Article 4 (December 2010)
71. Mühlen, M.Z., Becker, J.: Workflow management and object-orientation - A matter of perspectives or why perspectives matter. In: OOPSLA Workshop on Object-Oriented Workflow Management, Denver, USA (1999)
72. Nossum, A., Krogstie, J.: Integrated Quality of Models and Quality of Maps. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 264–276. Springer, Heidelberg (2009)
73. Nysetvold, A.G., Krogstie, J.: Assessing Business Process Modeling Languages Using a Generic Quality Framework. In: Siau, K. (ed.) Advanced Topics in Database Research, vol. 5, pp. 79–93. Idea Group, Hershey (2006)
74. Olle, T.W., Hagelstein, J., MacDonald, I.G., Rolland, C., Sol, H.G., van Assche, F.J.M., Verrijn-Stuart, A.A.: Information Systems Methodologies. Addison-Wesley (1988)
75. OMG, Semantics of Business Vocabulary and Rules Interim Specification (2006b), `http://www.omg.org/cgi-bin/doc?dtc/06/03/02` (retrieved January 1, 2006)
76. OMG. BPMN v2 Specification. Technical report, OMG (January 2011) (http://www.omg.org/), `http://www.omg.org/spec/BPMN/2.0/`
77. Ould, M.A.: Business Processes - Modeling and Analysis for Re-engineering and Improvement. Wiley, Beverly Hills (1995)

78. Pesic, M., van der Aalst, W.M.P.: A Declarative Approach for Flexible Business Processes Management. In: Eder, J., Dustdar, S. (eds.) BPM Workshops 2006. LNCS, vol. 4103, pp. 169–180. Springer, Heidelberg (2006)
79. Petri, C.A.: Kommunikation mit automaten (in German). Schriften des Rheinisch-Westfalischen Institut fur Instrumentelle Mathematik an der Universität Bonn (2) (1962)
80. Recker, J., Rosemann, M., Krogstie, J.: Ontology- versus pattern-based evaluation of process modeling language: A comparison. Communications of the Association for Information Systems 20, 774–799 (2007)
81. Scheer, A.-W., Nüttgens, M.: ARIS Architecture and Reference Models for Business Process Management, pp. 301–304 (2000)
82. Searle, J.R.: Speech Acts. Cambridge University Press (1969)
83. Searle, J.R.: Expression and Meaning. Cambridge University Press (1979)
84. Searle, J.R., Vanderveken, D.: Foundations of Illocutionary Logic. Cambridge University Press (1985)
85. Senge, P.: The Fifth Discipline: The Art and Practice of the Learning Organization. Century Business Publishers, London (1990)
86. Singh, B., Rein, G.L.: Role Interaction Nets (RINs); A Process Description Formalism, Technical Report CT-083-92, MCC, Austin, Texas (1992)
87. Stachowiak, H.: Allgemeine Modelltheorie. Springer, Wien (1973)
88. Suchman, L.: Do categories have politics? CSCW 2(3) (1994)
89. Swenson, K.D., et al.: A business process environment supporting collaborative planning. Journal of Collaborative Computing 1(1) (1994)
90. ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N. (eds.): Modern Business Process Automation: YAWL and its Support Environment. Springer (2010)
91. UMM - UN/CEFACT Modeling Methodology User Guide (2007)
92. Ward, P.T.: The transformation schema: An extension of the dataflow diagram to represent control and timing. IEEE Transactions on SE 12(2), 198–210 (1986)
93. Wegner, P., Goldin, D.: Interaction as a Framework for Modeling. In: Chen, P.P., Akoka, J., Kangassalu, H., Thalheim, B. (eds.) Conceptual Modeling. LNCS, vol. 1565, pp. 243–257. Springer, Heidelberg (1999)
94. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer-Verlag New York Inc. (2007)
95. WfMC Workflow Handbook 2001. Workflow Management Coalition, Future Strategies Inc., Lighthouse Point, Florida, USA (2000)
96. White. S.A. Introduction to BPMN. IBM Cooperation (2004)
97. Winograd, T., Flores, F.: Understanding Computers and Cognition. A-W (1986)