Oliver Thomas, Thorsten Dollmann, Peter Loos

Rules Integration in Business Process Models

A Fuzzy Oriented Approach

In business process management, decision situations are often characterized by fuzziness. This means that the decision premises are not available in the form of mathematic models or numeric values, but rather as fuzzy conditions, such as "low processing time" or "high quality". This article will show how fuzzy conditions and vaguely formulated goals in business process models can be considered using the fuzzy set theory. This fuzzy extension of process modelling is carried out with the event-driven process chain.

1 Fuzziness in Business Process Management

The goals in business engineering projects are the design of business processes and the analysis of requirements for their IT-support with regard to corporate strategies ([Sche94], [Meel94], [Tayl95], [VVVS98], [Dick06], [Öste07]). Process design must follow a comprehensive approach, comprising planning and control, i.e. the management of operational processes [BeKa03]. Modelling has proved helpful in supporting systematic procedures in process design. Modelling languages like the event-driven process chain (EPC) [KeNS92] serve as an operationalised approach to model construction. Software tools for business process modelling support the business engineer by giving him system components for the analysis, design and simulation of business process models [BlSi06].

Many concepts that consider situation-specific problems have been developed for the collection and improvement of business processes ([MaSc89], [Hamm90], [Robs91], [Dave93], [DuAH05], [DTHS07]), their generalization in reference models ([FeLo038], [ThSc06]) and their enterprise-specific adaptation in customizing ([BrBu06], [RoAa07], [DeKn07]).Many of these approaches focus on theuser-friendly and intuitive usability of thinking. More important however, for making the required decisions are the exact quantification and formalization of decision rules. Nevertheless, in many cases, only uncertain, imprecise and vague information about the often not technically determined procedures is available for business processes ([ReTu96], [HeLo99], [VöWe00], [VöWe02], [TALL06]). By the same token, the underlying goal system for process design is usually characterized by imprecise formulations and implicit interdependencies. An example for this is the statement "the processing time for orders with the priority 'very high' should be lowered 'considerably' while retaining a 'high' processing quality by 'adequately' reducing processing intensity". In this example, neither the concrete specification of both of the said goals regarding the processing time and quality, nor the measures derived from it can be quantified without loss of information and thus, operationalised. Information models, especially reference models, as well as methods for their enterprise-specific adaptation do not consider these forms of fuzziness as they should.

This shortcoming will be met here by extending process modelling through the consideration and processing of fuzziness with the Fuzzy-Set-Theory. This fuzzy extension will be reproduced with the EPC. The EPC was chosen as a process modelling language due significantly to its popularity in modelling practice. Our extension is however, not limited to the EPC or related modelling languages, whereby the latter term refers

This paper is based on Thomas, O.; Dollmann, Th.; Loos, P.: Towards Enhanced Business Process Models Based on Fuzzy Attributes and Rules. In: Proceedings of the 13th Americas Conference on Information Systems, AMCIS 2007, Paper 52.

to languages that, like the EPC for example, have no formal semantics or follow the paradigm of structured system development. The approach used here can also be transferred to object-oriented modelling languages (for example: UML-activity diagram). We will describe the steps and tools required for the extension as follows: first, we will specify the term "fuzziness" and motivate the consideration of fuzzy data using the fuzzy set theory. Then, the EPC will be introduced as a modelling language, formally defined and extended by the language constructs necessary for fuzzification. The introduction of the fuzzy-EPC, based on an attributation of EPC-language constructs, takes place in the next Section. After that, we will present an application scenario for our concept. The article ends with an analysis of related work and a discussion of our results.

2 From Crisp to Fuzzy Sets

There is no standard definition for the term "fuzziness" in literature – it seems as if the understanding of the term itself must remain fuzzy. Fuzziness is usually defined by way of differentiation with deterministic, stochastic and uncertain states of information ([DuPr97], [Zimm01], [Klir05]). In this article, fuzziness is seen as uncertainty with regard to data and its interdependencies. Different reasons for fuzziness can be identified in the business context (cp. Figure 1).

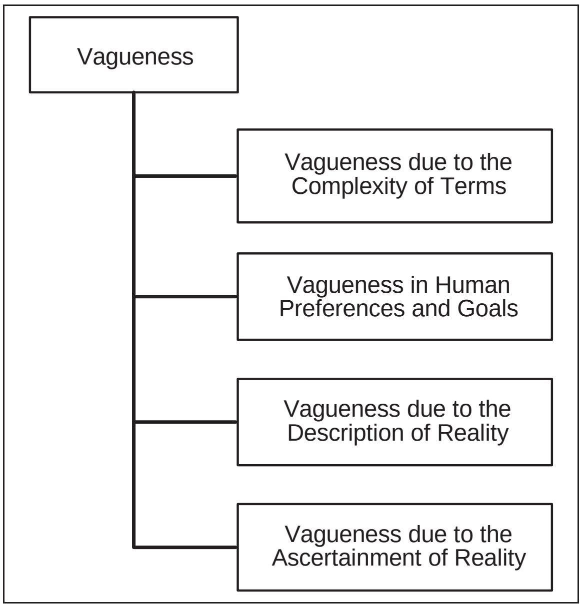


Figure 1: Fuzzy aspects

First, fuzziness occurs due to the complexity of the environment and the limits in human perception when comprehending reality. The resulting informational fuzziness, determined by human language and thought, can be ascribed to a surplus of information ([ZALW93], [Gali06]). This happens when terms with a high level of abstraction are used (for example: "credit worthiness"). Thus for example, knowledge intensive processes contain short-lived information from a number of sources, which results in the fact that only one part of the total process can be covered at one point in time. This part however, already becomes dated during the coverage of other sub-aspects. Many different attributes must be considered for the description of such complex terms. Fuzziness occurs because often, one is not capable of processing all of the relevant information and because, perhaps even the individual pieces of information themselves are already fuzzy. The descriptive attributes of the term are aggregated according to human information processing using linguistic terms.

Fuzziness also exists in human preference and goal conceptions. In many situations, human preference orders cannot be exactly determined. This leads to vagueness in the goal system, which is related to the informational fuzziness. For example, the goal "significant reduction in processing time" implicates measures. Often however, no action can be taken because of the inexplicit extent of the intended change and vague interdependencies with other goals.

The description of reality in natural language generates intrinsic (also: linguistic) fuzziness. The creation of a linguistic model and the context sensitivity of linguistic statements contribute to the creation of this fuzziness. The inaccuracy in linguistic comparisons is closely connected with this. An example for this is the statement "the object value is much higher than x". Here, the cause of fuzziness is not in the language itself, but rather in the limitation and subjectivity of human reality perception. The subjective conception of a circumstance by the person describing is expressed using language and no uniform definition for the terms used for the description exists.

Fuzziness when comprehending reality results from the fact that data and relationships between data can't or shouldn't be recorded exactly. The use of inaccurate data can also be advantageous when suitable measuring methods are lacking, the real-world is characterized by high dynamics or dependencies exist that cannot be determined accurately. Humans tend to register reality with verbal descriptions, which is another reason for the intrinsic fuzziness described above.

The fuzzy set theory attempts to overcome the separation between the necessary model and procedural precision on the one hand and the empirically desirable consideration of qualitative information on the other hand and to tolerate a portion of the precision that is lacking, as well as the vagueness and uncertainty in modelling processes.

The fuzzy set theory, today a branch of soft computing, was developed in the middle of the sixties [Zade65]. The core of the fuzzy theory is not to exclusively judge the status (of objects) with "true" or "false", but rather to allow intermediate stages. Based on Zadeh´s idea, the classical set theory is extended by the descriptions and linking of fuzzy sets. For each element ω of a given (crisp) basic set Ω the grade of membership to a subset $A \in \Omega$ is expressed through the value $\mu_A(\omega)$ of a mapping $\mu_A : \Omega \rightarrow [0;1]$. μ_A is called the "membership function" of the fuzzy set. The higher the membership grade of an element regarding a (fuzzy) set is, the more it belongs to this set.

Linguistic variables can be formulated with fuzzy sets [Zade73], which take on expressions in natural language – so-called linguistic terms – as values. Figure 2 shows the linguistic variable "order value". It has the terms "low", "medium" and "high". The membership of an object value to these fuzzy sets is expressed by the membership functions μ_{low}, μ_{medium} and μ_{high}. The object value 70,000 € belongs for example, to 0.5 to the fuzzy-set "medium", as well as to the fuzzy set "high". In a crisp context, it is only possible for example, to characterize an object value up from 70,000 € as a "high" order value, while 69,999 € would already pass for "medium".

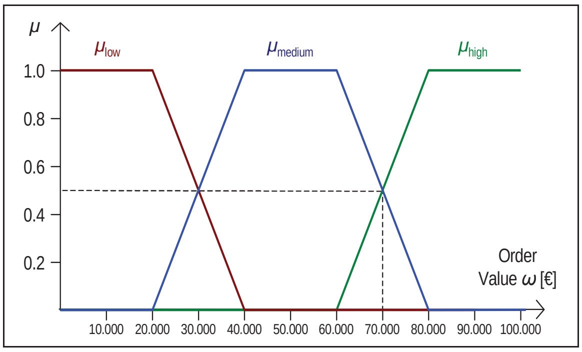


Figure 2: Linguistic variable "order value"

A fuzzy system has a fixed set of input and output variables, whose respective terms are connected with fuzzy rules consisting of a condition and a conclusion part, for example "WHEN customer assessment = middle AND order value = very high THEN order assessment = high". The value domains of the (linguistic) variables are partitioned by fuzzy sets, which serve the representation of the linguistic terms. A fuzzy rule can be represented formally as $(\mu^{(1)},\ldots,\mu^{(n)},\nu)$. $\mu^{(1)},\ldots,\mu^{(n)}$ are fuzzy sets over the value domain of the input variables and ν is a fuzzy set over the value domain of the output variables. The input and output variables are assigned to each other by inference mechanisms. If $X = X_1 \times \cdots \times X_n$ is the input space and Y the output space, then a fuzzy system FS can be formally represented as a mapping $FS : X \rightarrow Y$ [BKNK03].

The fuzzy rule base defines the structure of the fuzzy systems. Based on a vector of input entities $\vec{x} = (x_1,\ldots,x_n) \in X$, the (crisp) default value of a typical fuzzy system $y = FS(\vec{x})$ can be calculated in several steps. First, the degree of performance for each individual rule is found by determining the value of the grade of membership to the corresponding fuzzy set. Then the corresponding grades of membership must be connected conjunctively with a suitable fuzzy operator. Several fuzzy sets result from each individual rule. These must be combined disjunctively for the determination of the output of the fuzzy system. A crisp value for the output variable is required for an executable action, for example "determine priority". A defuzzification step delivers this crisp value $y \in Y$ from the output fuzzy set.

If the output variable is not a continuous entity, but rather a categorical variable that can take on any discrete values (classes), then one speaks of a classification problem. A rule-based classification can be modelled with a fuzzy system by understanding each class as a special fuzzy set and selecting the class with the highest grade of membership as a default value for the fuzzy system in a defuzzification step.

3 Process Modelling with the EPC

3.1 The Modelling Language EPC

Since the establishment of the process idea for the organization of businesses and the design of information systems, a large number of modelling languages for the description of business processes has been used [DuAH05]. The EPC has established itself for the construction of business process models on a conceptual level because of its application orientation and comprehensive tool support, especially in the German-speaking community.

In graph-theoretical terminology an EPC-model is an ordered and connected graph, whose nodes are events, functions and logical connectors. Events are the passive elements in the EPC. They describe the arrival of a certain state und are represented by hexagons (for example: "Customer order (is) defined", cp. Figure 3). Functions, represented by rounded rectangles, are the active elements in the EPC. The term "function" is equated with a task in the EPC. Events trigger functions and they are their result. Control flow edges, symbolized by arrows, represent these

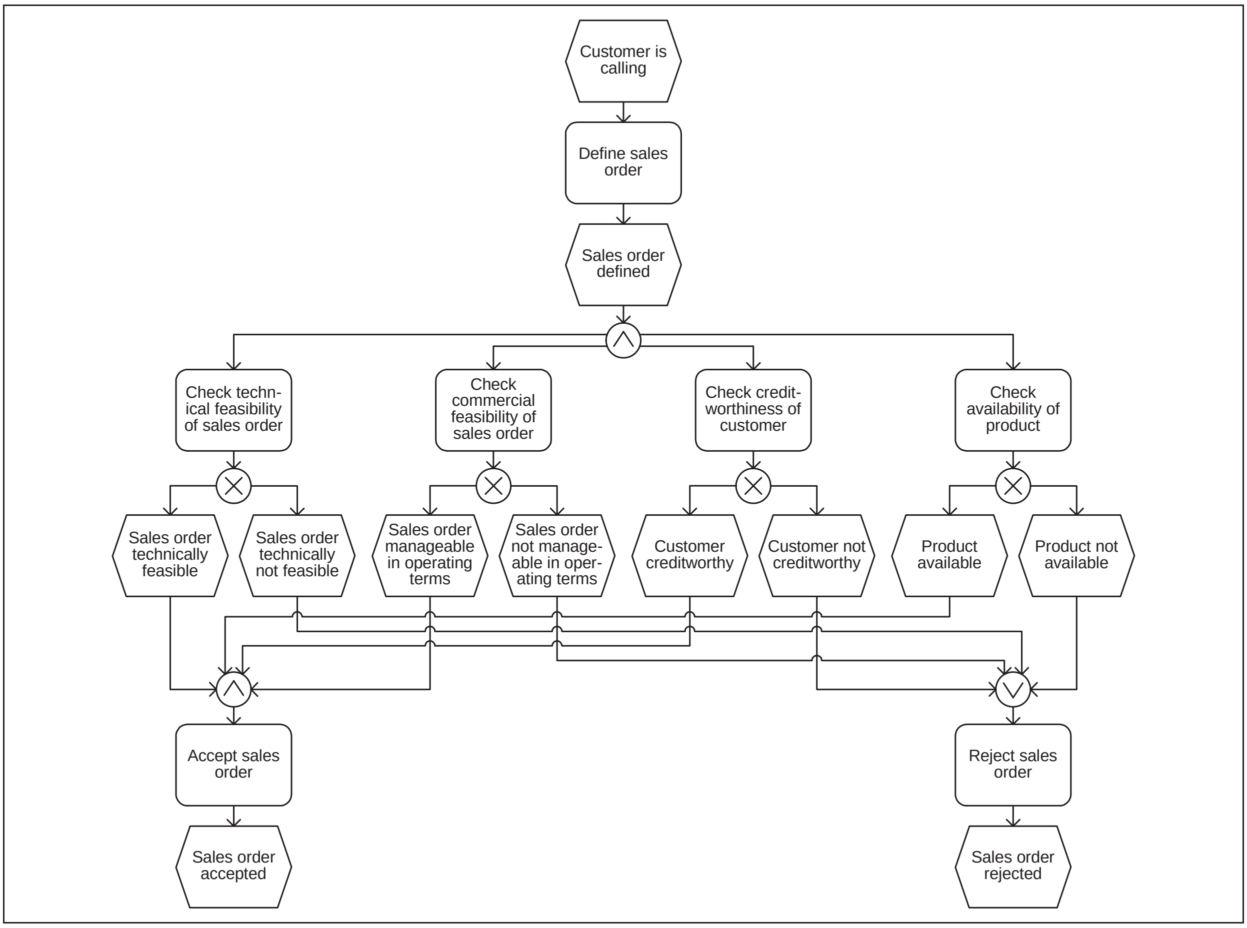


Figure 3: An EPC-model for customer order processing

relationships between functions and events. Logical connectors are used to express the fact that a function is started by one or more events and can generate one or more events as a result respectively. One differentiates, in accordance with the terminology of propositional logic, between conjunctive, adjunctive and disjunctive logical connectors (cp. Figure 3). The corresponding connectors are referred to simply as AND, OR resp. XOR-connectors.

With this information, the following interpretation results for the process model in Figure 3: the model describes the procedure for the definition and execution of test functions for a customer order. The decision as to whether a customer order is accepted or rejected is made through the parallel execution of various subfunctions. The sales order is checked with reference to technical feasibility and from a business view. In addition, customer creditworthiness and product availability are checked. Negative results, such as for example “Sales order not technically feasible” or “Customer not creditworthy” lead to the rejection of the customer order through the function “Reject sales order”.

3.2 Formalization of the EPC

A formal definition of the syntax and semantics of models is necessary for consistency checks or the automated processing of EPC-models, for example, in software tools for simulation or verification. Various approaches to a formal syntax and semantics definition for the EPC have been suggested and discussed in academic circles ([Aals99], [AaDK02], [Kind04], [Kind06], [RoAa07]). In the following, we will introduce a formal definition for the syntax of EPC-models according to Rosemann and van der Aalst [RoAa07], in order to then based upon this, precisely define a fuzzy extension. The resulting set-theoretical specification does not serve to illustrate behavioural aspects of EPC-models. Based on the original definition of EPC-models, semantic ambiguities especially arise in the use of the OR-connector, because its control behaviour is not always locally determinable. Due to this, its use is discussed in literature ([LaSW98], [DeRi01]). A joining OR-connector can synchronize or not, i. e. it can operate or however, wait for several events after it receives the first input. This ambiguity must be accommodated for by increased technical coordination between the creator of the model and its user, whereby the further processing of the model must occur in reference to the context. We end the semantic discussion regarding the control flow semantics of the EPC-method at this point and refer to the articles mentioned.

In formal notation, an EPC-model is a quadruple $EPC = (E, F, C, A)$. E is thereby a finite (non-empty) set of events, F a finite (non-empty) set of functions, $C = C_{AND} \cup C_{OR} \cup C_{XOR}$ a finite set of logical connectors, whereby C_{AND}, C_{OR} and C_{XOR} are paired disjunctive subsets of C and

$$A \subseteq (E \times F) \cup (F \times E) \cup (E \times C) \cup (C \times E) \cup (F \times C) \cup (C \times F) \cup (C \times C)$$

is a set of edges. The relation A specifies the set of ordered control flow edges (arcs), which connect functions, events and connectors with each other. $V = E \cup F \cup C$ is the set of all nodes of the EPC-model.

To introduce the concept of the syntactic correctness of EPC-models, we define the set of input nodes of a node $v \in V$ with $\bullet v := \{w \in V \mid (w, v) \in A\}$ and the set of its corresponding output nodes with $v\bullet := \{w \in V \mid (v, w) \in A\}$ for an EPC-model. Furthermore, we write an ordered path from a node $v_1 \in V$ to a node $v_l \in V$ ($l \in \mathbb{N}$) as a sequence $p = \langle v_1, \ldots, v_l \rangle$ of nodes $v_i \in V$ with $(v_i, v_{i+1}) \in A$, whereby $1 \leq i \leq l-1$. We define

$$C_{XY} := \left\{ \begin{array}{l} c \in \{v_2, \ldots, v_{l-1}\} \\ \mid \exists \text{ path } p = \langle v_1, v_2, \ldots, v_{l-1}, v_l \rangle \\ \text{with } v_1 \in X \wedge v_2, \ldots, v_{l-1} \in C \wedge v_l \in Y \end{array} \right\}$$

for the set of tangent connectors on a path between nodes from the sets $X, Y \in \{E, F\}$.

On the syntactic level, some rules have established themselves and serve the construction of syntactically correct EPC-models [RoAa07]. With them, the consistency of an EPC-model can be checked. To do so, an EPC-model $EPC = (E, F, C, A)$ must fulfil the following conditions:

- (V, A) is an ordered and connected graph. This means that no isolated objects exist in EPC-models.
- Events have at most one incoming and at most one outgoing edge: $\forall e \in E : |\bullet e| \leq 1 \wedge |e \bullet| \leq 1$.
- Functions have exactly one incoming and exactly one outgoing control flow edge: $\forall f \in F : |\bullet f| = 1 \wedge |f \bullet| = 1$.
- There is at least one start and one end event: $\exists e \in E : |\bullet e| = 0 \wedge \exists e \in E : |e \bullet| = 0$.
- Connectors either have several incoming and one outgoing control flow edge (join) or one incoming and several outgoing control flow edges (split): $\forall c \in C : (|\bullet c| = 1 \wedge |c \bullet| > 1) \vee (|\bullet c| > 1 \wedge |c \bullet| = 1)$.
  We refer to $C_J = \{c \in C \mid |\bullet c| > 1\}$ as the set of join-connectors and with $C_S = \{c \in C \mid |c \bullet| > 1\}$ the set of split-connectors and $C_J \cap C_S = \emptyset$ holds.
- The graph expanded by the EPC-model is simple, i. e. it contains no loop (edge with the same start and end nodes) and no multiple edges between the individual nodes: $\forall v_1, v_2 \in V$:
  $$\left| \{a \in A \mid a = (v_1, v_2) \vee a = (v_2, v_1)\} \right| \leq 1.$$
- There is no ordered circle in the EPC graph, which consists only of logical connectors: $C = C_{EF} \cup C_{FE}$ and $C_{EF} \cap C_{FE} = \emptyset$, i. e. the sets C_{EF} and C_{FE} partition C and a connector $c \in C$ lies either on a path from an event to a function or on a path from a function to an event.
- Functions are only connected with events (if necessary, over logical connectors) and vice versa: $C_{EE} = \emptyset \wedge C_{FF} = \emptyset$.
- After events, no XOR or OR-split-connector follows in the control flow: $C_S \cap C_{EF} \cap C_{XOR} = \emptyset = C_S \cap C_{EF} \cap C_{OR}$.

The last rule ensures the exclusion of decision making with events in accordance with the approved types of connectors. We adhere to this important principle in the fuzzy extension and rule out a determined branch of process flows after events. The consistency of an EPC-model can be checked using the rules mentioned above. In the following, we will continue to speak of EPC-models and in doing so, always refer to the set of the syntactically correct EPC-models according to the known rules.

Further statements result through the use of the EPC as a central modelling language for the architecture of integrated information systems (ARIS) [Sche99].

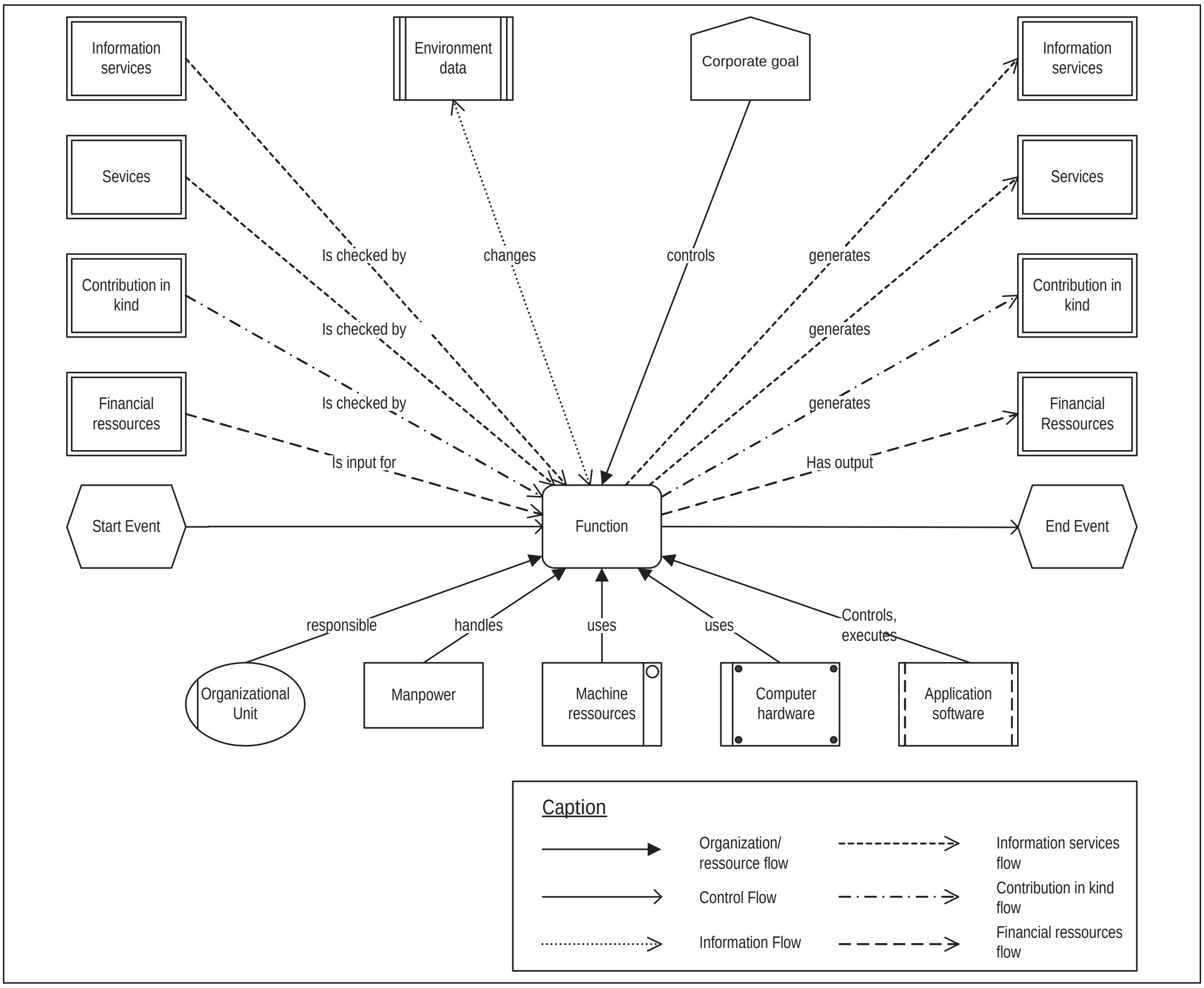


Figure 4: Extension of the EPC with ARIS-language constructs [Sche98, 31]

These are based on the ARIS-view concept. They are made through the annotation of other language constructs on EPC-functions [ScTA05]. Thus, for example, language constructs that represent the environment data, news, manpower, machine resources and computer hardware, application software, outputs in the form of contributions in kind, services and information services, financial resources, organizational units or corporate goals are recommended (cp. Figure 4).

The linkage of constructs that can only take place with functions from the EPC is created with edges, which, in addition to the control flow already introduced, can be differentiated in organization or resource, information, information services and contribution in kind, as well as financial resources flow [Sche98, 31].

In this article, we chose the EPC elements of the organization, data and output view as additional artefacts for process modelling, added them to the formal representation of the EPC and, in a next step, enriched them with attributes. This extension will be consulted later for the demonstration of the exemplary processing of fuzziness in business processes.

To do this, we introduce an EPC-model extended by ARIS-language constructs as a tuple

$$EPC_{ARIS} = (E, F, C, A, O, D, L, R).$$

(E, F, C, A) is an EPC-model with the set of control flow nodes $V = E \cup F \cup C$ and the set of control flow edges A. The node set, which represents the artefacts of the organization, data and output view, are O for the set of organizational units, D for the set of data objects and L for the set of outputs. It is required that the sets O, D and L are pairwise disjoint. The set R contains sets of relations, which assign the functions the various artefacts. The individual relations have different meanings and define the type of relation between the elements from $O \times F$,...

, $F \times L$. Exemplary practical relation types are "is input for" or "is responsible for".

An EPC-model extended with ARIS-language constructs $EPC_{ARIS} = (E, F, C, A, O, D, L, R)$ is then syntactically correct when (E, F, C, A) is a syntactically correct EPC-model and in addition, each artefact is connected with at least one node from the EPC-graph (V, A) , whereby only annotated artefacts on functions are allowed here. Thus, we postulate that the ARIS-extended graph $G = (V_A, A \cup R)$ with the set of nodes $V_A = E \cup F \cup C \cup O \cup D \cup L$ and the set of edges $A \cup R$ is connected.

4 Fuzzy-Event-Driven Process Chain

4.1 Extension of the EPC with attributes

The object types in EPC-models (for example: the individual data objects from D or organizational units from O), understood as object sets of individual objects - one also speaks of *instances* of the respective type[1] - are characterized by certain attributes. These characteristics are used on the one hand, to describe the individual objects and on the other, for their internal representation, for example, for storage in relational databases and are referred to as *attributes*. While descriptive attributes represent technical characteristics, so-called *key attributes* serve the clear-cut identification of an object. A customer can for example, be identified with his name, address and date of birth, while his volume of sales or customer assessment represent application-relevant characteristics. In the following, only economically relevant attributes will be consulted and considered in the fuzziness-concept.

Each attribute has a value domain, which defines the set of possible attribute values. For instance, the value domain for the attribute "Order value" of a data object type "Order" can be defined as a set of natural numbers. Similarly, the value set for the attribute "Name" of the object type "Customer" can be defined to the set of character strings composed of alphabetical characters.

In most cases, one abstains from representing attributes in conceptual EPC-models due to reasons of clearness and complexity. However, the following fuzzy extension requires a specification of the conceptual model and thus, an explicit modelling of decision-relevant attributes in the modelling process.

S is a set of objects, $Dom(A_i)$ ($i = 1, \ldots, n$), $n \in \mathbb{N}$, sets of values and A_i ($i = 1, \ldots, n$) well-defined mappings of the form

$$A_i : S \to Dom(A_i),\ (i = 1, \ldots, n).$$

Then $\{A_i, \ldots, A_n\}$ is a set of attributes on the objects of the set S or on S . The set $Dom(A_i)$ is called the value domain of the attribute A_i and the elements $A_i(s) \in Dom(A_i)$ are called the attributes of the object s . If $Dom(A_i) = \{0,1\}$ applies then A_i is called the binary attribute on S . Under the premises of this definition we have an (internal) representation of the objects as the tuple $(A_1(s), \ldots, A_n(s))$ of attribute values, i. e. as elements of the set

$$Dom(A_i) \times \cdots \times Dom(A_n) = \prod_{i=1}^{n} Dom(A_i) .$$

In an ARIS-EPC-model we create attributes on events, functions, organizational units, data objects and outputs.

Thus, in a conceptual EPC-model designed on the type level each node element in the EPC-graph is assigned its own attributes. This is made clear for example, by the fact that a data object(-type) "Sales order" has an attribute "Order size", whereas this attribute does not represent an attribute of a data object "Article".

We define an ARIS-EPC-model extended with attributes as a tuple

$$EPC_{ARIS,attr} = (E, F, C, A, O, D, L, R, M) .$$

The individual events from E , functions from F , organizational units from O , data objects from D and input from L are thereby assigned to attributes. At this point, we will do without the assignment of attributes for the set of control flow edges A and the relations from R , because these descriptive attributes are not consulted for fuzzification. The listed attributes from elements from E, F, O, D and L are combined in the set M .

Each object has its own identifying and application-relevant attributes with their own value sets. Only attributes relevant in the respective context are modelled. Changes in the attributes of the artefacts are therefore, only considered as far as it is apparent from the EPC-model.

4.2 Fuzzy-extension of the EPC

We define a Fuzzy-EPC-model

$$FEPC = (E, F, C, A, O, D, L, R, M, FC)$$

1. Up to now, we have done without the difference between the type and instance level in process models. At this point, we will be more exact and speak of object types in EPC-models and their instances. The function type "Check customer credit worthiness" as an element of the set F can for example, generate any number of instances during the model's runtime.

as an ARIS-EPC-model enriched with attributes with the following properties:

- M is the set of fuzzy attributes of the Fuzzy-EPC-model *FEPC*. The term "fuzzy attribute" refers here to two aspects. First, one assumes that the value domains of the attributes are not necessarily crisp sets, but rather may consist of fuzzy sets. And second, the attributes can be interpreted as linguistic variables. This implies that the name of the linguistic variable corresponds with the name of the attribute and that the value domain of the attribute is, at the same time, the basic set of the linguistic variable.

- O, D and L are sets of organizational units, data objects resp. outputs, which contain the fuzzy organizational units, fuzzy data objects and fuzzy outputs. Here, fuzzy organizational units, fuzzy data objects and fuzzy output are organizational units, data objects and output with fuzzy attributes.

- FC is a set of fuzzy systems. The possible input and output quantities are restricted by the function assigned to such a system.

- F is the set of fuzzy functions of the EPC-model. A fuzzy function is characterized here by either one or more fuzzy attributes or by the assignment of a fuzzy system $FS \in FC$ for decision support on the basis of fuzzy formulated rules during process execution. Thereby all of the organizational units, data objects and outputs of the EPC-model, whose attributes represent the input and output quantities of the assigned fuzzy system, must be connected with this fuzzy function via an edge. If the fuzzy system is used directly as a classificator for the decision on the further control flow, then only the following events of this function may occur in the conclusion part of the rules. In this case, the fuzzy classification system FS can be formally represented as a mapping

- $FS : \prod_{A_i \in A, i \in \{1,\ldots,n\}} Dom(A_i) \to \mathrm{P}(E)$, where $\mathrm{P}(E)$ is the power set of the set of events.

The detail of differentiation between fuzzy systems and fuzzy classification systems is also motivated through the following fact: while "fuzziness" could be passed on through fuzzy systems to other process functions which can use it, fuzzy classification systems lead to a crisp decision in the process flow.

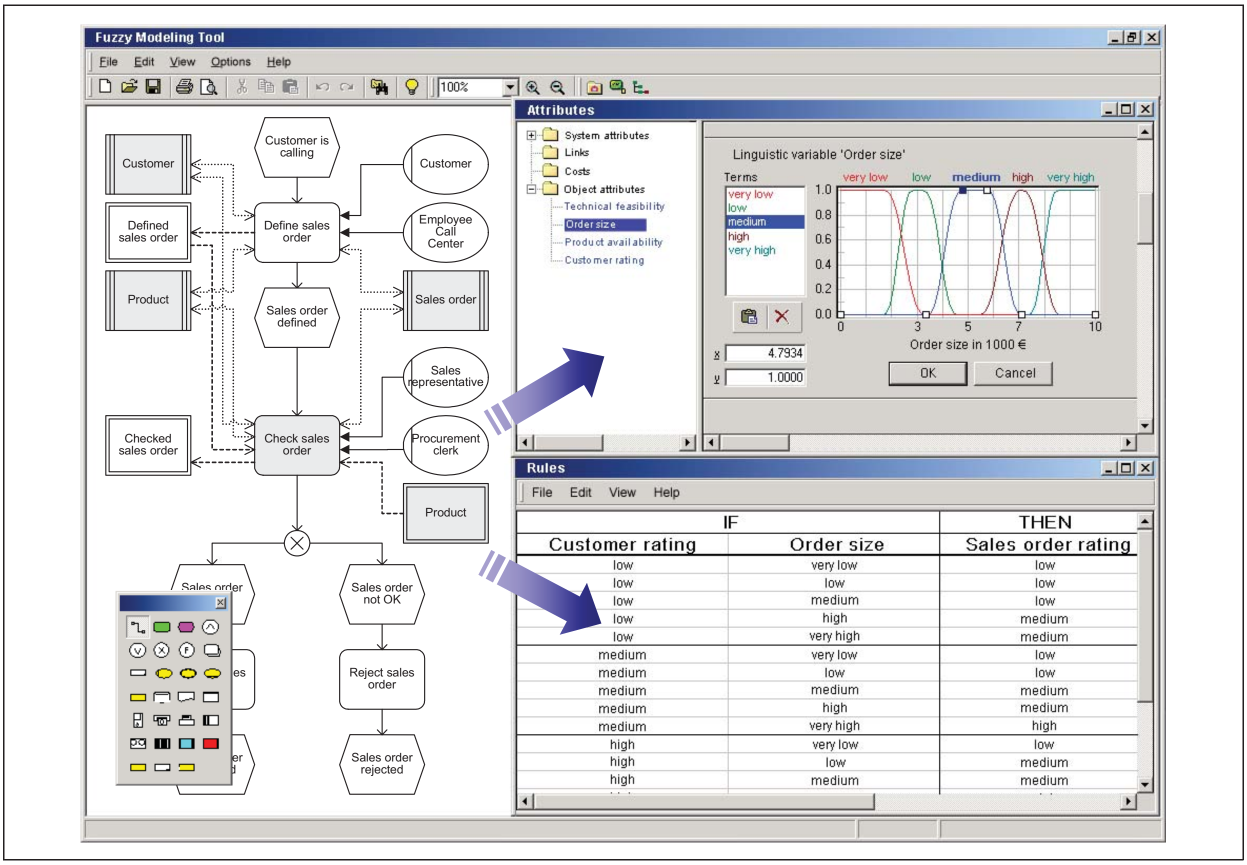
The fuzzy-extension of the event-driven process chain will be demonstrated in the following Section based on an application scenario.

5 Application Scenario "Fuzzy Customizing"

The construction of process models is often connected with the demand to abstract from enterprise-specific attributes in order to make the models reusable. Therefore, one differentiates between enterprise-specific process models and reference process models. The term "enterprise-specific" characterizes the individual character of a respective model. In contrast to this, a reference model represents a point of reference for the development of specific models, because it stands for a class of applications ([Thom06], [RoAa07]). Prominent examples are the reference model for industrial business processes (Y-CIM-Model) from Scheer [Sche94], as well as the Supply-Chain Operations Reference-model [SCC08].

Figure 3 shows a section of a reference process for customer order processing in the form of an EPC. A weak point in the modelling process not yet discussed is recognizable here: each of the negative results leads to the immediate rejection of the customer order – irrespective of the check results from the other functions. This is contradictory to business practice where such absolute elimination criteria are only rarely complied. In fact, decision-maker use implicit compensation mechanisms, which counter-balance an exceedance of limiting values in one area with better values in another area. The rules for the interdependent impact are not documented here, but rather based upon the decision-makers know-how. Furthermore, it is usually a case of simple rules, which establish only scale-related combinations and which orient themselves on target systems with vague interdependences.

In the present case, the decision as to whether a product is available could be answered not only with a crisp "yes" or "no", but rather also be characterized by the additional effort resulting from weighing things up, so that the product could, for example be requested from another warehouse, if all other inspections turned out to be positive. This results in the challenge of representing fuzziness in reference and procedure models for their adaptation, in addition to the problem of the development of implicit knowledge. Figure 5 shows the fuzzy extension of the reference process for sales order processing – embedded in the graphic user interface of a fuzzy modelling tool. The process is represented in the main window in the form of a Fuzzy-EPC. The fuzzy constructs of the EPC are characterized by grey shading. After defining the



Customer rating	Order size	Sales order rating
low	very low	low
low	low	low
low	medium	low
low	high	medium
low	very high	medium
medium	very low	low
medium	low	low
medium	medium	medium
medium	high	medium
medium	very high	high
high	very low	low
high	low	medium
high	medium	medium

Figure 5: Interface of the fuzzy-modelling tool

customer order, its acceptance is checked. The checking of the individual functions in the "crisp" processes is however, extended by way of inspections pertaining to the size of the order and customer appraisal. The functions are not modelled as "subordinate" activities of the customer order check, but rather as fuzzy object attributes of the respective data object and input types in the form of linguistic variables (cp. Figure 5, Window "Attributes"). In the attribute-explorer for example, the object attribute "Order size" of the data object type "Sales order" is activated. It has the linguistic variables "very low", "low", "medium", "high" and "very high" as terms (cp. also Figure 5).

On the right side of the attribute window the user can change the membership functions of the linguistic terms with a variable editor. A variable assistant supports the user by way of an automated variable definition. A rule editor (cp. same window in Figure 5) shows the rules for the function. In the example, a rule set with the input variables "Customer rating" and "Order size", as well as the output variables "Customer order rating" is given. The user creates the rule sets in the table by for example, the automated adoption of complete rule sets from a rule assistant equipped with "consistency checks" (interface to the fuzzy system)

The reference process consists – in accordance with the formalization of the Fuzzy-EPC introduced – of two levels in its extension. The modelling level (cp. Figure 5, left) still shows the process model, in this case a Fuzzy-EPC-model. On this level, the semi-formal modelling is limited to the content necessary for end-users to understand the business logic. In a further level (cp. Figure 5, right), the decision-supporting rules are shown, which cause the acceptance or rejection of a customer order. This level uses knowledge from the fuzzy-set-theory to represent the characteristics of calculative decisions.

The adaptation of such a process is now limited to the expert knowledge stored in the decision rules and does not affect the process logic of the process. By considering fuzzy conditions and vaguely formulated objectives with the help of approaches from the fuzzy set theory, the user with expert knowledge can carry out the adaptation of the reference model himself with intuitive and simple linguistic evaluations.

Nevertheless, it must be mentioned that an evaluation of the fuzzy-extended EPC is connected with the application scenario “Fuzzy Customizing”. In fact, the application at hand serves to show that new requirements for technical built-time-modelling result from the EPC-language extension. When designing the process model, one must decide which situations can now be described using rules from decision logic that, up to now, were mapped in the crisp process logic of the process model. Thus, as shown in our example, the design procedure is changed for technical models, as well as the construction results.

6 Related Work

There are few approaches that integrate fuzziness aspects in information resp. process modelling with the Fuzzy-Set-Theory.

The fuzzy-extension of Entity-Relationship-Models (ERM) was described by Zvieli, Chen [ZvCh86]. Here, types of entities, relations and attribute sets can take on fuzzy-values. The consideration of these fuzzified data structures consequently leads to the processing of fuzzy data in the respective business processes. Fuzzy theory-based extensions of object-oriented modelling methods for business processes can be found in [BSVV98] and ([Cox99], [Cox02]). An object-oriented approach based on the Fuzzy-Set-Theory for the simulation of business processes is presented by Völkner, Werners [VöWe02]. Among others, Petri nets are used for the description of the dynamic aspects information systems. The bivalent behaviour of places and transitions in a Petri net is however, a disadvantage when mapping knowledge intensive and weakly structured processes. In order to represent the system behaviour with fuzzy process conditions or incomplete, vague information, Petri-Nets were extended by fuzzy-concepts. The Fuzzy Petri net [GüLi93] is created by the projection of several crisp Petri nets, in which the structural information is mapped as fuzzy sets. Rehfeldt, Turowski ([ReTu96], [ReTu98]) demonstrate the consideration of fuzzy data in business process modelling with the event-driven process chain on the example of industrial order processing. Vague sales information is seen as important input data and transformed into tentative customer orders. This “fuzzy extension” of the process is visualized by shaded objects. From a methodical view, fuzzy and crisp model objects must not be differentiated in the conceptual representation of a business process. Moreover, rules and parameters relevant for process execution should be mentioned in early stages of process design. Thomas and Adam [AdTh05] examine, in cooperation with other co-authors, how fuzzy data can be used for the design of knowledge-intensive and weakly structured business processes and how their implementation can be used in application systems.

The existing works on fuzzy EPCs have two essential similarities: First, they deal with modelling aspects as used in Business Process Reengineering or for the introduction of ERP systems, i. e. aspects, which apply to the build-time of the process models. Second, the point in all of them is to embed the approaches successful in fuzzy logic for the control and regulation of the decision situations relevant for company processes. Thus, the latter applies to the runtime for process models. Independent of the methodological basis of the existing works, almost all of the authors target the integration of two classes of tools: on the one hand, process modelling tools and on the other, fuzzy systems. In doing so, their integration must be supported by a suitable information technical design. Still, despite the many studies available, up to now no formalization exists for the Fuzzy EPC and was thus addressed in this article.

7 Summary and Outlook

An approach for the integration of fuzzy aspects in business process management was developed in this article. The integration was carried out in two ways. First, the fuzzy data was considered with the help of the Fuzzy-Set-Theory. Second, it was carried out on the example of an established modelling language for business processes, the Event-driven Process Chain. The concept corresponds in a figurative sense with a “level extension” of the language: while the business process models are limited to the content necessary for the end user to understand the business logic, the expert knowledge is stored for the decision-support of individual model elements.

It was shown in the applications described that many situations in business process management could be described more exactly through the modelling of vague knowledge with fuzzy logic. Therefore, rule-based systems founded on fuzzy logic are well suited for controlling processes. Because the rule base is based on IF-THEN-rules, its functional behaviour can be understood relatively easily and existing knowledge can be integrated simply. This makes the constant improvement of the process definitions in the sense of a continuous process improvement easier.

The authors see a future challenge for their research above all in answering the question, as to whether the creation of adequate linguistic variables and rule bases can occur economically in fuzzy-business process management. Setting up a rule base proves to be especially problematic in practice. The developer must analyze each undesired malfunction and correct it by hand. By optimizing rule-based fuzzy-systems with neural networks, fuzzy sets can be adapted and the rule base learned resp. corrected. The capability of artificial neural networks to uncover business logic in processes ("Process Mining"), as well as to improve business processes through learning are currently being discussed. A combination with fuzzy logic in this context is seen as a promising field [TiTM08].

In this study, we considered vague information in business process with fuzzy logic. This allowed an adequate representation of operational processes and improved decision support in the execution of business processes. However, the services, activities and data objects in business process management are also subject to fuzzy interpretation. This takes place during model construction when a business engineer formulates the aspects of a business process that are relevant for him in a process model. In doing so, he assigns identifiers, used for communication with the model users, to the model elements. The model users then interpret the identifiers in the models and assign corresponding terms or facts to them. Due to the fuzziness inherent in natural language the benefits of semiformal process models as a medium for communication between model designers and users are limited. Further problems result with regard to the practicability of the process models. This is generally not immediately clear, but rather requires yet another interpretation during implementation activities.

8 Acknowledgements

This work has been funded by the State Ministry of Education and Research of Saarland in the research project "Fuzzy Process Engineering" and the *Deutsche Forschungsgemeinschaft* (German Research Foundation) as part of the research project "Fuzzy-Customizing". The authors would like to thank the anonymous reviewers for their comments which helped to improve an earlier version of this paper.

References

[AaDK02] van der Aalst, W. M. P.; Desel, J.; Kindler, E.: On the semantics of EPCs: A vicious circle. In: Nüttgens, M.; Rump, F. J. (eds.): EPK 2002 : Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, Proceedings. Bonn : GI, 2002, pp. 71-79

[Aals99] van der Aalst, W. M. P.: Formalization and verification of event-driven process chains. In: Information and Software Technology 41 (1999) 10, pp. 639-650

[AdTh05] Adam, O.; Thomas, O.: A Fuzzy Based Approach to the Improvement of Business Processes. In: Castellanos, M.; Weijters, T. (eds.): BPI'05 : Workshop on Business Process Intelligence ; Nancy, France, September 5, 2005, pp. 25-35

[BeKa03] Becker, J.; Kahn, D.: The Process in Focus. In: Becker, J.; Kugeler, M.; Rosemann, M. (eds.): Process Management : A Guide for the Design of Business Processes. Berlin : Springer, 2003, pp. 1-11

[BKNK03] Borgelt, C.; Kruse, R.; Nauck, D.; Klawonn, F.: Neuro-Fuzzy-Systeme : Von den Grundlagen künstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen. 3rd ed. Wiesbaden : Vieweg, 2003 (in German)

[BlSi06] Blechar, M. J.; Sinur, J.: Magic Quadrant for Business Process Analysis Tools, 2006. Stamford, CT : Gartner Research, 2006

[BrBu06] vom Brocke, J.; Buddendick, C.: Reusable Conceptual Models - Requirements Based on the Design Science Research Paradigm. In: Hevner, A. R. (ed.): First International Conference on Design Science Research in Information Systems and Technology : February 24-25, 2006, Claremont ; Proceedings, 2006

[BSVV98] Benedicenti, L.; Succi, G.; Vernazza, T.; Valerio, A.: Object Oriented Process Modeling with Fuzzy Logic. In: Carroll, J.; Lamong, G. B.; Oppenheim, D.; George, K. M.; Bryant, B. (eds.): Applied computing 1998 : Proceedings of the 1998 ACM Symposium on Applied Computing ; Atlanta, Georgia, February 27-March 1, 1998. Danvers : ACM Press, 1998, pp. 267-271

[Cox02] Cox, E.: Knowledge-Based Business Process Modeling: Complex Systems Design Through A Fusion of Computational Intelligence And Object-Oriented Models. In: PC AI 16 (2002) 2, pp. 15-23

[Cox99] Cox, E.: Striving for Imprecision: Fuzzy Knowledge Bases for Business Process Modeling. In: PC AI 13 (1999) 4

[Dave93] Davenport, T. H.: Process Innovation : Reengineering Work through Information Technology. Boston : Harvard Business School Press, 1993

[DeKn07] Delfmann, P.; Knackstedt, R.: Towards Tool Support for Information Model Variant Management - A Design Science Approach. In: Österle, H.; Schelp, J.; Winter, R. (eds.): Proceedings of the Fifteenth European Conference on Information Systems. University of St. Gallen, 2007, pp. 2098-2109

[DeRi01] Dehnert, J.; Rittgen, P.: Relaxed Soundness of Business Processes. In: Dittrich, K. R.; Geppert, A.; Norrie, M. C. (eds.): Proc. of the 13th Conference on Advanced Information Systems Engineering (CAISE'01), Interlaken, Switzerland. Berlin : Springer, 2001, pp. 157-170

[Dick06] Dickinson, B.: Strategic business engineering : a synergy of software engineering and information engineering. Brisbane : LCI Press, 2006

[DTHS07] Dalmaris, P.; Tsui, E.; Hall, B.; Smith, B.: A framework for the improvement of knowledge-intensive business processes. In: Business Process Management Journal 13 (2007) 2, pp. 279-305

[DuAH05] Dumas, M.; van der Aalst, W. M. P.; ter Hofstede, A. H. M. (eds.): Process-aware Information Systems : Bridging People and Software through Process Technology. Hoboken, New Jersey : Wiley, 2005

[DuPr97] Dubois, D.; Prade, H. M.: The three semantics of fuzzy sets. In: Fuzzy Sets and Systems 90 (1997), pp. 141-150

[FeLo03] Fettke, P.; Loos, P.: Classification of Reference Models - A Methodology and its Application. In: Information Systems and e-Business Management 1 (2003) 1, pp. 35-53

[Gali06] Galindo, J.; Urrutia, A.; Piattini, M. (eds.): Fuzzy databases: modeling, design, and implementation. Hershey, Pa. Idea Group Publ., 2006

[GüLi93] Günther, R.; Lipp, H.-P.: A Fuzzy Petri Net Concept for Complex Decision Making Processes in Production Control. In: Zimmermann, H.-D. (ed.): Proceedings of the 1st European Congress on Fuzzy and Intelligent Technologies, EUFIT '93 : Aachen, Germany, 7.-10. September 1993. Aachen : Augustinus, 1993, pp. 290-295

[Hamm90] Hammer, M.: Reengineering Work: Don't Automate, Obliterate. In: Harvard Business Review 68 (1990) 4, pp. 104-112

[HeLo99] Herrmann, T.; Loser, K.-U.: Vagueness in models of socio-technical systems. In: Behavior and Information Technology 18 (1999) 5, pp. 313-323

[KeNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". In: Scheer, A.-W. (ed.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, no. 89, Saarbrücken : Universität des Saarlandes, 1992 (in German)

[Kind04] Kindler, E.: On the Semantics of EPCs : A Framework for Resolving the Vicious Circle. In: Desel, J.; Pernici, B.; Weske, M. (eds.): Business Process Management : Second International Conference (BPM 2004), Potsdam, Germany, June 17-18, 2004; Proceedings. Springer, 2004, pp. 82-97

[Kind06] Kindler, E.: On the semantics of EPCs: Resolving the vicious circle. In: Data & Knowledge Engineering 56 (2006) 1, pp. 23-40

[Klir05] Klir, G. J.: Uncertainty and Information : Foundations of Generalized Information Theory. Hoboken : Wiley-Interscience, 2005

[LaSW98] Langner, P.; Schneider, C.; Wehler, J.: Petri Net Based Certification of Event driven Process Chains. In: Desel, J.; Silva, M. (eds.): Application and theory of Petri nets 1998 : 19th international conference ; proceedings. Berlin : Springer, 1998, pp. 286-305

[MaSc89] Mansir, B. E.; Schacht, N. R.: Continuous Improvement Process : Principles and Practices. Springfield : NTIS, 1989

[Meel94] Meel, J. W. v.: The dynamics of business engineering : reflections on two case studies within the Amsterdam Municipal Police Force. Delft University of Technology, doctoral dissertation, 1994

[Öste07] Österle, H.: Business Engineering: Transition to the Networked Enterprise. In: Electronic Markets 6 (2007) 2, pp. 14-16

[ReTu96] Rehfeldt, M.; Turowski, K.: A Tool-supported Distributed Application of Fuzzy Logic in Order Processing. In: Jamshidi, M.; Yuh Junku; Dauchez, P. (eds.): Proceedings of the World Automation Congress (WAC '96) : May 28-30, 1996, Montpellier, France : Intelligent automation and control : Recent trends in development and applications. Albuquerque : TSI Press, 1996, pp. 585-589

[ReTu96] Rehfeldt, M.; Turowski, K.: Fuzzy Objects in Production Planning and Control. In: Zimmermann, H.-J. (ed.): Proceedings / EUFIT '96, Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September 2-5, 1996 / ELITE Foundation. Aachen : Mainz, Wissenschaftsverlag, 1996, pp. 1985-1989

[ReTu98] Rehfeldt, M.; Turowski, K.: A Flexible Java-based Fuzzy Kernel for Business Applications. In: Alpaydin, E. (ed.): International ICSC symposium on engineering of intelligent systems : EIS'98. ICSC Academic Press, 1998, pp. 204-209

[RoAa07] Rosemann, M.; van der Aalst, W. M. P.: A configurable reference modelling language. In: Information Systems 32 (2007) 1, pp. 1-23

[Robs91] Robson, G. D.: Continuous process improvement : simplifying work flow systems. New York : Free Press, 1991

[SCC08] Supply-Chain Council (ed.): Supply-Chain Operations Reference-model, SCOR Overview, Version 9.0, SCC, Washington DC, 2008

[Sche94] Scheer, A.-W.: Business Process Engineering : Reference Models for Industrial Enterprises. 2nd ed. Berlin : Springer, 1994

[Sche98] Scheer, A.-W.: ARIS - business process frameworks. 2nd ed. Berlin : Springer, 1998

[Sche99] Scheer, A.-W.: ARIS - business process modeling. 2nd ed. Berlin : Springer, 1999

[ScTA05] Scheer, A.-W.; Thomas, O.; Adam, O.: Process Modeling Using Event-driven Process Chains. In: Dumas, M.; van der Aalst, W. M. P.; ter Hofstede, A. H. M. (eds.): Process-aware Information Systems : Bridging People and Software through Process Technology. Hoboken : Wiley, 2005, pp. 119-145

[TALL06] Thomas, O.; Adam, O.; Leyking, K.; Loos, P.: A Fuzzy Paradigm Approach for Business Process Intelligence. In: IEEE Joint Conference on E-Commerce Technology (CEC 2006) and Enterprise Computing, E-Commerce and E-Services (EEE 2006) : June 26-29, 2006, San Francisco, California. Los Alamitos : IEEE Computer Society Press, 2006, pp. 206-213

[Tayl95] Taylor, D. A.: Business engineering with object technology. New York : Wiley, 1995

[Thom06] Thomas, O.: Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation. In: Bussler, C.; Haller, A. (eds.): Business Process Management Workshops : BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM, WSCOBPM, BPS, Nancy, France, September 5, 2005 ; Revised Selected Papers. Berlin : Springer, 2006, pp. 484-496

[ThSc06] Thomas, O.; Scheer, A.-W.: Tool Support for the Collaborative Design of Reference Models - A Business Engineering Perspective. In: Sprague, R. H. (ed.): Proceedings of the 39th Annual Hawaii International Conference on System Sciences : 4-7 January 2006, Kauai, Hawaii ; Abstracts and CD-ROM of Full Papers. Los Alamitos : IEEE Computer Society Press, 2006

[TiTM08] Tiwari, A.; Turner, C. J.; Majeed, B.: A review of business process mining: state-of-the-art and future trends. In: Business Process Management Journal 14 (2008) 1, pp. 5-22

[VöWe00] Völkner, P.; Werners, B.: A decision support system for business process planning. In: European Journal of Operational Research 125 (2000) 3, pp. 633-647

[VöWe02] Völkner, P.; Werners, B.: A simulation-based decision support system for business process planning. In: Fuzzy Sets and Systems 125 (2002) 3, pp. 275-288

[VVVS98] de Vreede, G.-J.; Vogel, D.; Verbraeck, A.; Sol, H. G.: The Dynamics of Business Engineering. In: 31st Annual Hawaii International Conference on System Sciences, Kohala Coast, Hawaii, USA, January 6-9, 1998 : Volume 1: Collaboration Systems and Technology. IEEE Computer Society, 1998, pp. 632

[Zade65] Zadeh, L. A.: Fuzzy sets. In: Information and Control 8 (1965) 3, pp. 338-353

[Zade73] Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes. In: IEEE Transactions on Systems, Man and Cybernetics 3 (1973) 1, pp. 24-44

[ZALW93] Zimmermann, H.-J.; Angenstenberger, J.; Lieven, K.; Weber, R. (eds.): Fuzzy-Technologien : Prinzipien, Werkzeuge, Potentiale. Düsseldorf : VDI, 1993 (in German)

[Zimm01] Zimmermann, H.-J.: Fuzzy set theory - and its applications. 4th ed. Boston : Kluwer Acad. Publ., 2001

[ZvCh86] Zvieli, A.; Chen, P. P.-S.: Entity-Relationship Modeling and Fuzzy Databases. In: International Conference on Data Engineering, February 5-7, 1986, Bonaventure Hotel, Los Angeles, California, USA. Washington, DC : IEEE Computer Society Press, 1986, pp. 320-327

Oliver Thomas, Thorsten Dollmann, Peter Loos

Institute for Information Systems (IWi)
at the German Research Center
for Artificial Intelligence (DFKI),
Saarland University
Campus D3 2, Stuhlsatzenhausweg 3
66123 Saarbruecken
Germany
{oliver.thomas|thorsten.dollmann|peter.loos}@iwi.dfki.de