Towards Enhanced Business Process Models Based on Fuzzy Attributes and Rules

Oliver Thomas, Thorsten Dollmann, Peter Loos
Institute for Information Systems (IWi)
at the German Research Center for Artificial Intelligence (DFKI),
Saarland University, Saarbruecken (Germany)
[oliver.thomas|thorsten.dollmann|peter.loos]@iwi.dfki.de

ABSTRACT

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 modeling is carried out with the event-driven process chain.

Keywords

Business Process Management, Business Process Automation, Business Process Modeling, Business Rules, Event-Driven Process Chains, Fuzzy Set Theory

FUZZINESS IN BUSINESS PROCESS MANAGEMENT

The goals in current business engineering projects lay in the design of business processes and the analysis of requirements for their IT-support with regard to corporate strategies (Österle & Winter, 2003). Process design must follow a comprehensive approach, comprising planning and control, i.e. the management of operational processes (Becker, Kugeler, & Rosemann, 2005). Modeling has proved to be helpful for the support of systematic procedures in process design. Modeling languages like the event-driven process chain (EPC) (Keller, Nüttgens, & Scheer, 1992) serve as an operationalized approach to model construction. Software tools for business process modeling (Blechar & Sinur, 2006) support the business engineer with system components for the analysis, design and simulation of business process models.

Many concepts that consider situation-specific problems have been developed for the collection and improvement of business processes, their generalization in reference models and their enterprise-specific adaptation in customizing. Many of these approaches focus on the user-friendly and intuitive usability of methods by modeling them on human ways of thinking. More important for making the required decisions are however, the exact quantification and formalization of decision rules. However, in many cases, only uncertain, imprecise and vague information about the often not technically determined procedures is available for business processes (Völkner, 1998; Forte, 2002; Hüsselmann, 2003). 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, operationalized. Information models, especially reference models, as well as methods for their enterprise-specific adaptation still don't consider these forms of fuzziness as they should.

This circumstance will be met here by extending process modeling through the consideration and processing of fuzziness using the fuzzy-set-theory. This fuzzy extension will be reproduced with the EPC. The EPC was chosen as a process modeling language due significantly to its popularity in modeling practice. Our extension is however, not limited to the EPC or "related" modeling languages, whereby the latter term refers to languages that, like the EPC for example, have no formal semantics or also follow the paradigm of structured system development. The approach presented can also be transferred to object-oriented modeling languages (for example: UML-activity diagram) or modeling languages with formal semantics (for example: Petri-nets). 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 modeling 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 similar research in and a discussion of our results.

FROM CRISP TO FUZZY SETS

There is no standard definition for the term "fuzziness" in literature – it almost 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 (Rehfeldt, 1998, S. 39). 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 (Tietze, 1999, S. 45 ff.) (cp. Fig. 1).

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 (Zimmermann et al., 1993, S. 5). 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, man is not capable of processing all of the relevant information and because, perhaps even the individual pieces of information are themselves already fuzzy. The descriptive attributes of the term are aggregated according to human information processing using linguistic terms.

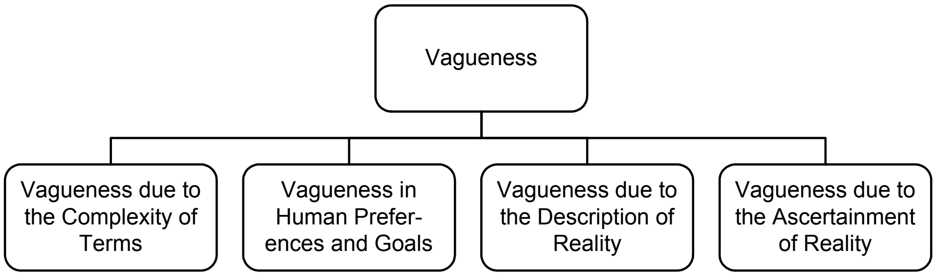


Fig. 1. Fuzzy aspects

Fuzziness also exists in human preference and goal conceptions. In many situations, preferences human preference orders cannot be determined exactly. This leads to a 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: verbal or 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 human reality perception (Tietze, 1999, S. 47). 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 however, 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 modeling processes.

The fuzzy set theory, today a branch of soft computing, was developed in the middle of the sixties (Zadeh, 1965). 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 original idea, the classical set theory, i. e. the theory of crisp sets, 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 $\mu_A : \Omega \to [0;1]$. These grades of membership are selected from the interval $[0;1]$ and the following interpretation results: the higher the membership grade of an element regarding a (fuzzy) set is, the more it belongs to this set. μ_A is called the "membership function" of the fuzzy set $\{(\omega, \mu_A(\omega)) \mid \omega \in \Omega\}$.

Linguistic variables can be formulated with fuzzy sets (Zadeh, 1973), which take on expressions in natural language – so-called linguistic terms – as values. Fig. 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". This mapping of crisp values on fuzzy values is called "fuzzification". In a crisp context, it is only be 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".

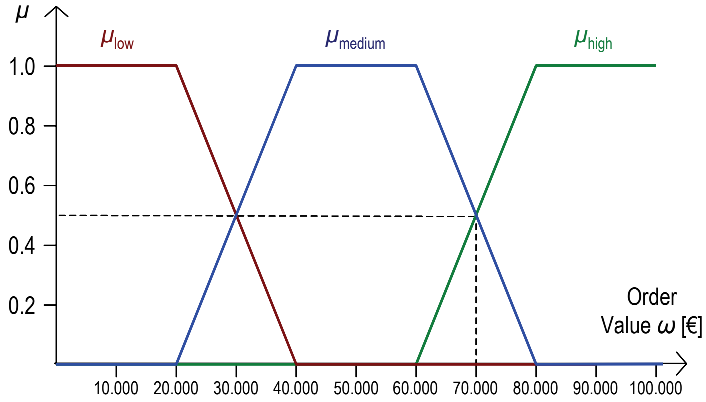


Fig. 2. Linguistic variable "Order value"

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 $FS : X \to Y$ (Borgelt et al., 2003, S. 162f).

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. From each individual rule result several fuzzy sets. 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 categorial variable that can take on any discrete values (classes), then one speaks of a classification problem. A rule-based classification can be modeled 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.

PROCESS MODELING WITH THE EPC

The Modeling Language EPC

Since the establishment of the process idea for the organization of businesses and the design of information systems, a large number of modeling languages for the description of business processes has been used (Dumas, van der Aalst, & ter Hofstede, 2005). 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. It was developed at the Institute for Information Systems (IWi) at the Saarland University in Saarbruecken, in cooperation with SAP, Inc. (Keller, Nüttgens, & Scheer, 1992).

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. Functions, represented by rounded rectangles, are the active elements in the EPC. The term "function" is equated with a task in the EPC. An event refers to a point in time, in contrast to a function, which is a time-consuming happening. While in literature the suggestion is made (e.g. Hoffmann, Kirsch, & Scheer, 1992, S. 5) to use the respective object of the processing and an infinitive verb for labeling the function, i.e. task, to be carried out (for example: "Define customer order", cp. Fig. 3). The recommendation for events is to pair the object experiencing the change in state with a verb in present perfect that describes the type of change (for example: "Customer order (is) defined", cp. Fig. 3).

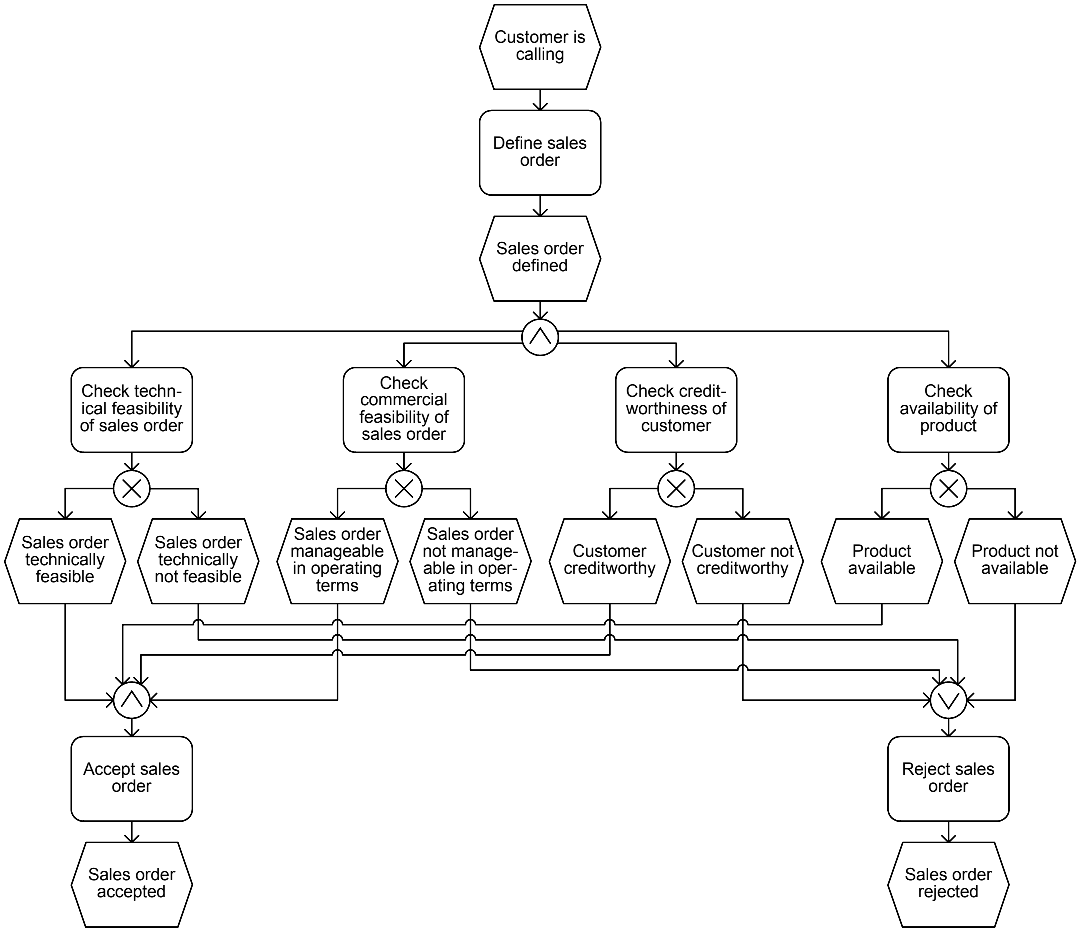


Fig. 3. An EPC-model for customer order processing

Events trigger functions and are their result. Control flow edges, symbolized by arrows, represent these relationships between functions and events. Logical connectors are used to express that a function is started by one or more events resp. that a function can generate one or more events as a result. One differentiates, in accordance with the terminology of propositional logic, between conjunctive "∧", adjunctive "∨" and disjunctive logical connectors "⊗" (cp. Fig. 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 Fig. 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 sub-functions. The customer 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 "Customer order not technically feasible" or "Customer not creditworthy" lead to the rejection of the customer order through the function "Reject customer order".

Formalization of the EPC

The notation for the event-driven process chain introduced by Keller, Nüttgens and Scheer (Keller, Nüttgens, & Scheer, 1992) was initially developed as a non-formalized notation and used without formal semantics. This suffices for documenting processes and using models as a basis for discussion. However, 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 (van der Aalst, 1999; Nüttgens & Rump, 2002; van der Aalst, Desel, & Kindler, 2002; Kindler, 2004; Kindler, 2006; Rosemann & van der Aalst, 2006). In the following, we will introduce a formal definition for the syntax of EPC-models according to Rosemann and van der Aalst (Rosemann & van der Aalst, 2006), in order to then based upon this, precisely define a fuzzy extension. The resulting set-theoretical specification does not serve to illustrate behavioral 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 behavior is not always locally determinable. Due to this, its use is discussed in literature (Langner, Schneider, & Wehler, 1998; Rittgen, 2000; Dehnert & Rittgen, 2001). 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. Kindler shows that the semantics cannot always be clearly defined due to the non-locality of the Join-OR and XOR-connector (Kindler, 2006). 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.

However, it is exactly the use of the XOR and OR-connectors that increases freedom of expression in EPC-models and that is in part responsible for the success of the EPC in practice, because the semi-formal character represents a balance between formal accuracy and an intuitive usability for the expert. According to popular modeling practice one assumes that a Join-OR-connector is activated when the first input enters the system. 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 connects 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 $\bullet v := \{w \in V \,|\, (w, v) \in A\}$ and the set of its corresponding output nodes with $v \bullet := \{w \in V \,|\, (v, w) \in A\}$ 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

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

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 (Keller & Teufel, 1999, 172–174; Nüttgens & Rump, 2002, 68–70). With them, the consistency of an EPC-model can be checked. For this, an EPC-model $EPC = (E, F, C, A)$ must fulfill the following conditions:

- (V, A) with $V = E \cup F \cup C$ is an ordered and connected graph.[1]
- 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: $\exists e \in E : |\bullet e| = 0 \wedge \exists e \in E : |e \bullet| = 0$.
- Logical connectors either have several incoming and one outgoing control flow edge (join) or one outgoing and several incoming 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 $C_J = \{c \in C \,||\bullet c| > 1\}$ as the

---

[1] At the same time, this conditions means that $\forall v \in V : |\bullet v| > 0 \vee |v \bullet| > 0$ holds, i.e. that no isolated objects exist in EPC-models.

set of join-connectors and with $C_S = \{c \in C \,||\, c \bullet | > 1\}$ the set of split-connectors of the EPC-model $EPC = (E, F, C, A)$ and $C_J \cap C_S = \varnothing$ holds.

- The graph expanded by the EPC-model is simple (irreflexive and antisymmetric = asymmetric), i.e. it contains no loop (edge with the same start and end nodes)[2] 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 $C_{EF} \cap C_{FE} = \varnothing$, 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: $C_{EE} = \varnothing \wedge C_{FF} = \varnothing$.
- After events, no XOR or OR-split-connector follows in the control flow: $C_S \cap C_{EF} \cap C_{XOR} = \varnothing$ and $C_S \cap C_{EF} \cap C_{OR} = \varnothing$.

The last rule ensures the exclusion of decision making with events in accordance with the approved types of connectors (Keller, Nüttgens, & Scheer, 1992). 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.

ARIS-extension of the EPC

Further statements result through the use of the EPC as a central modeling language for the architecture of integrated information systems (ARIS) (Scheer, 2001; Scheer, 2002). These are based on the ARIS-view concept. They are made through the annotation of other language constructs on EPC-functions (Scheer, Thomas, & Adam, 2005). 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. Fig. 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/resource, information, information services and contribution in kind, as well as financial resources flow (Scheer, 2002, S. 31).

In this article, we chose the EPC elements of the organization, data and output view as additional artifacts for process modeling (cp. Fig. 4), add them to the formal representation of the EPC and, in a next step, enrich them with attributes. This extension will be consulted later for the demonstration of the exemplary processing of fuzziness in business processes.

For 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 artifacts of the organization, data resp. 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 artifacts. It is defined as the set $R = R^{OF} \cup R^{DF} \cup R^{FD} \cup R^{LF} \cup R^{FL}$, whereby

- $R^{OF} = \{R_1^{OF}, \ldots, R_{n_{OF}}^{OF}\}$, with R_i^{OF} $(1 \leq i \leq n_{OF})$, $n_{OF} \in \mathbb{N}$, are relations on $O \times F$,
- $R^{DF} = \{R_1^{DF}, \ldots, R_{n_{DF}}^{DF}\}$, with R_i^{DF} $(1 \leq i \leq n_{DF})$, $n_{DF} \in \mathbb{N}$, are relations on $D \times F$,
- $R^{FD} = \{R_1^{FD}, \ldots, R_{n_{FD}}^{FD}\}$, with R_i^{FD} $(1 \leq i \leq n_{FD})$, $n_{FD} \in \mathbb{N}$, are relations on $F \times D$,
- $R^{LF} = \{R_1^{LF}, \ldots, R_{n_{LF}}^{LF}\}$, with R_i^{LF} $(1 \leq i \leq n_{LF})$, $n_{LF} \in \mathbb{N}$, are relations on $L \times F$ and
- $R^{FL} = \{R_1^{FL}, \ldots, R_{n_{FL}}^{FL}\}$, with R_i^{FL} $(1 \leq i \leq n_{FL})$, $n_{FL} \in \mathbb{N}$, are relations on $F \times L$.

---

[2] This requirement is fulfilled for $v \in E$ and $v \in F$ through the limitation of A to $A = V \times V \setminus \{(E \times E) \cup (F \times F)\}$. For $v \in C$ the constraint $(v, v) \notin A$ is implicitly guaranteed through the fact that the following applies, per definition: $\forall v \in C: (v \in C_{EF} \vee v \in C_{FE}) \wedge (v \in C_S \vee v \in C_J)$.

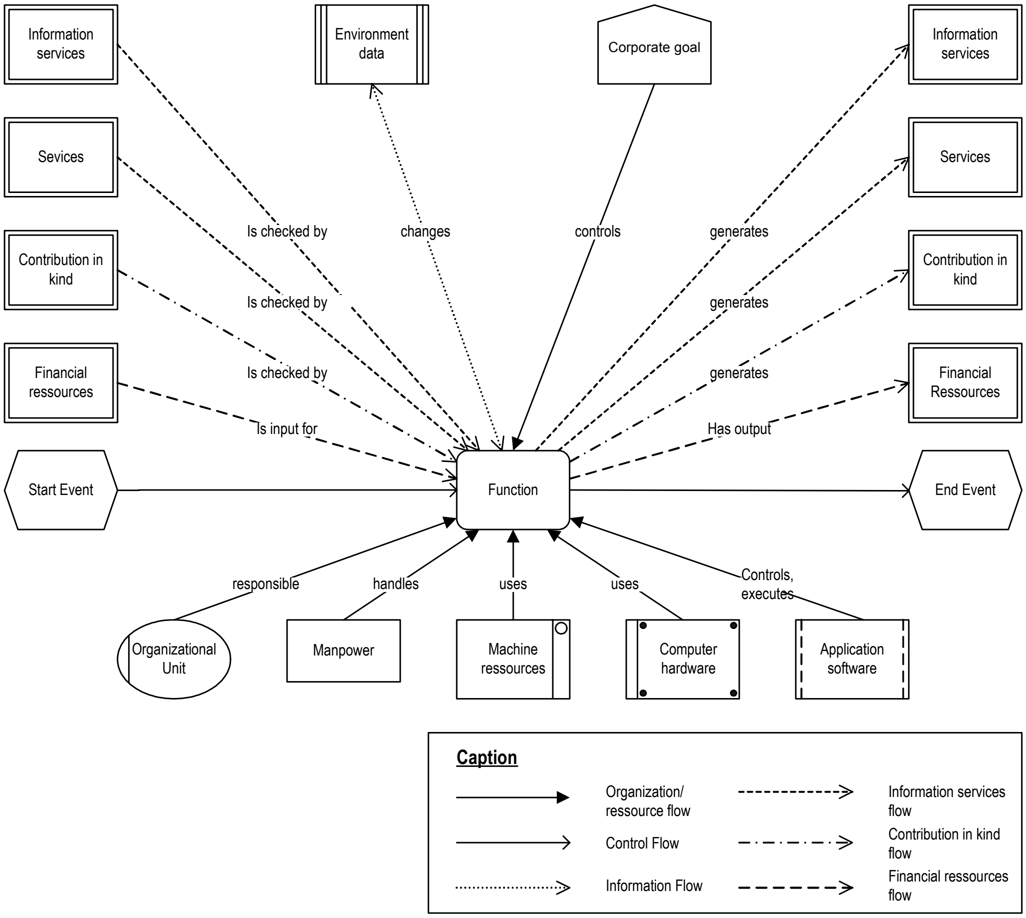


Fig. 4. Extension of the EPC with ARIS-language constructs (Scheer, 2002, S. 31)

The individual relations from the sets $R^{OF}, R^{DF}, R^{FD}, R^{LF}$ and R^{FL} have different meanings and define the type of relation between the elements from $O \times F, \ldots, F \times L$. A selection of practical relation types is listed in Table 1.

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 artifact is connected with at least one node from the EPC-graph (V, A), whereby only annotated artifacts on functions are allowed here. Thus, we postulate that 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.

Table 1. Types of relations between functions and ARIS-language constructs

Source object type	*Target object type*	*Possible types of relationships*
Organizational unit	Function	Carries out, decides about, is responsible for, agrees, cooperates in, must be informed, must inform about result from
Data object	Function	is input for, is approved by, is checked by
Function	Data object	changes, has output, generates
Output	Function	is input for, is consumed by, is used by
Function	Output	has output, produces

FUZZY-EVENT-DRIVEN PROCESS CHAIN

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[3], are characterized by certain attributes. These characteristics are used one 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 modeling of decision-relevant attributes in the modeling 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

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

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

$$Dom(A_1) \times \cdots \times Dom(A_n) = \prod_{i=1}^{n} Dom(A_i).$$

In an ARIS-EPC-model $EPC_{ARIS} = (E,F,C,A,O,D,L,R)$ we create the following attributes:

- $A_1^e,\ldots,A_{n_e}^e$ ($n_e \in \mathbb{N}$) are the n_e attributes on the event $e \in E$,
- $A_1^f,\ldots,A_{n_f}^f$ ($n_f \in \mathbb{N}$) are the n_f attributes on the function $f \in F$,
- $A_1^o,\ldots,A_{n_o}^o$ ($n_o \in \mathbb{N}$) are the n_o attributes on the organizational unit $o \in O$,
- $A_1^d,\ldots,A_{n_d}^d$ ($n_d \in \mathbb{N}$) are the n_d attributes on the data object $d \in D$,
- $A_1^l,\ldots,A_{n_l}^l$ ($n_l \in \mathbb{N}$) are the n_l attributes on the output $l \in L$.

Thus, in a conceptual EPC-model modeled 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) “Customer order” has an attribute “Order value”, 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 modeled. Changes in the attributes of the artifacts are therefore only considered, as far as it is apparent from the EPC-model.

---

[3] 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 instance during the models runtime.

Fuzzy-extension of the EPC

We define a Fuzzy-EPC-model

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

as an ARIS-EPC-model enriched with attributes $EPC_{ARIS,attr} = (E, F, C, A, O, D, L, R, M)$ 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 resp. L are sets of organizational units, data objects resp. outputs, which contain the fuzzy organizational units, fuzzy data objects resp. fuzzy outputs. A fuzzy organizational unit, a fuzzy data object resp. fuzzy output is here an organizational unit, a data object resp. 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 resp. 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.
- The set R contains sets of fuzzy relations[4] between control flow objects and the various artifacts $R = \{R^{OF}, R^{DF}, R^{FD}, R^{LF}, R^{FL}\}$, with
  - $R^{OF} = \{R_1^{OF}, \ldots, R_{n_{OF}}^{OF}\}$, whereby R_i^{OF} $(1 \le i \le n_{OF})$, $n_{OF} \in \mathbb{N}$, are fuzzy relations on $O \times F$,
  - $R^{DF} = \{R_1^{DF}, \ldots, R_{n_{DF}}^{DF}\}$, whereby R_i^{DF} $(1 \le i \le n_{DF})$, $n_{DF} \in \mathbb{N}$, are fuzzy relations on $D \times F$,
  - $R^{FD} = \{R_1^{FD}, \ldots, R_{n_{FD}}^{FD}\}$, whereby R_i^{FD} $(1 \le i \le n_{FD})$, $n_{FD} \in \mathbb{N}$, are fuzzy relations on $F \times D$,
  - $R^{LF} = \{R_1^{LF}, \ldots, R_{n_{LF}}^{LF}\}$, whereby R_i^{LF} $(1 \le i \le n_{LF})$, $n_{LF} \in \mathbb{N}$, are fuzzy relations on $L \times F$, and
  - $R^{FL} = \{R_1^{FL}, \ldots, R_{n_{FL}}^{FL}\}$, whereby R_i^{FL} $(1 \le i \le n_{FL})$, $n_{FL} \in \mathbb{N}$, are fuzzy relations on $F \times L$.

The relations in the crisp model can thus be seen as a special instance of the fuzzy case in the sense of Zadeh‘s extension principle. The fuzzy-extension of the event-driven process chain will be demonstrated in the following section based on a case study.

APPLICATION SCENARIO “FUZZY CUSTOMIZING”

Due to the possibility of their reutilization, in many cases the construction of process models is connected to the demand to abstract from enterprise-specific characteristics. Therefore, one differentiates between enterprise-specific process models und 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 (vom Brocke, 2003; Thomas, 2006). Prominent examples in the scientific field are the reference model for industrial business processes (Y-CIM-Model) from Scheer (1997), as well as the SAP R/3-reference model which is a result of business practice (Keller & Teufel, 1999).

Fig. 3 shows a section of a reference process for customer order processing in the form of an EPC. A weak point in the modeling 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, through man as the decision-maker implicit

---

[4] A fuzzy relation R over sets Ω_1, Ω_2 is a fuzzy set of the cartesian product $\Omega_1 \times \Omega_2$, which is characterized via a membership grade $\mu_R : \Omega_1 \times \Omega_2 \to [0,1]$. In it, each element (ω_1, ω_2) as a 2-digit tuple in R is assigned a membership grade $\mu_R(\omega_1, \omega_2) \in [0,1]$. The membership grade is interpreted as intensity of the fuzzy relation R between the elements of the tuple.

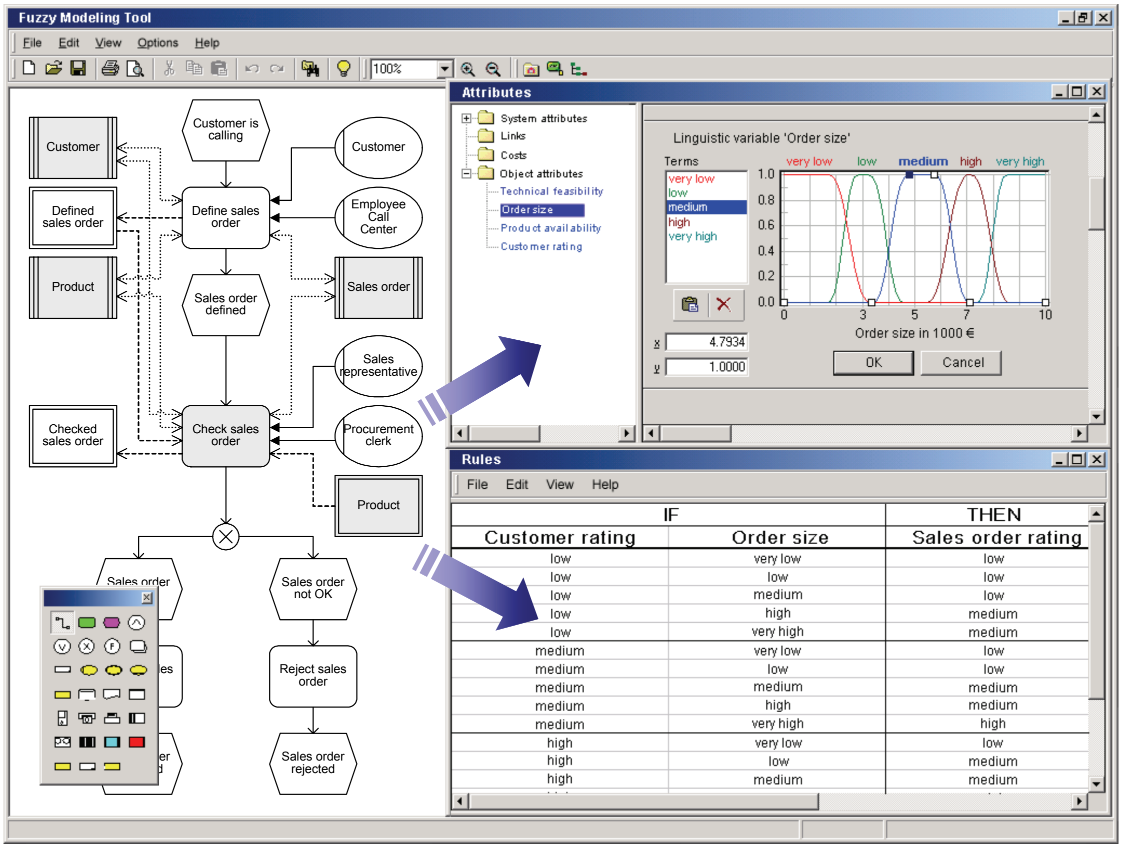
compensation mechanisms are used, which counter-balance an exceedance of limiting values in one area with better values in another area. The rules for the interdependent impact are not documented here, but rather based upon the decision-makers know-how. Furthermore, it is usually a case of simple rules, 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 for example, could be requested from another warehouse, if all other inspections turned out to be positive. A corresponding decision orients itself on the trade-off between the goal to avoid additional costs and the focus on customer needs. This results in the challenge to represent fuzziness in reference and procedure models for their adaptation, in addition to the problem of the development of implicit knowledge.

Fig. 5 shows the fuzzy extension of the reference process for customer order processing – embedded in the graphic user interface of a fuzzy modeling 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 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 modeled 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. Fig. 5, Window "Attribute"). In the attribute-explorer for example, the object attribute "Contract volume" of the data object type "Customer order" is activated. It has the linguistic variables "very low", "low", "medium", "high" and "very high" as terms (cp. also Fig. 5).

In the right part of the attribute window the user can change the membership functions of the linguistic terms with a variable editor, for example by "pulling" the "corners" of the functions symbolized by small rectangles. A variable assistant supports the user by way of an automated variable definition. A rule editor (cp. same window in Fig. 5) shows the rules for the function.



x 4.7934
y 1.0000

IF		THEN
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

Fig. 5. Interface of the fuzzy-modeling tool

In the example, a rule set with the input variables “Customer assessment” and “Contract volume”, as well as the output variables “Customer order assessment” 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 modeling level (cp. Fig. 5, left) still shows the process model, in this case a fuzzy-EPC-model. On this level, the semi-formal modeling is limited to the content necessary for end-users to understand the business logic. In a further level (cp. Fig. 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.

A further result is that an already adapted process can in principle, be understood as a reference process – the process logic of the process remains unchanged during its adaptation, it is only the type of decision-finding that must be adapted.

Nevertheless, it must be mentioned that an evaluation of the fuzzy-extended EPC is connected with the application scenario “Fuzzy Customizing”. A tool-supported simulation was already carried out at the Institute for Information Systems at the DFKI in Saarbruecken (Adam, Thomas, & Loos, 2006). In fact, the application at hand serves to show that new requirements for technical built-time-modeling result from the EPC-language extension. When designing the process model, one must decided 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.

RELATED WORK

There are few approaches, which integrate fuzziness aspects in information resp. process modeling with the Fuzzy-Set-Theory.

The fuzzy-extension of Entity-Relationship-Models (ERM) was described by Zvieli, Chen (1986). 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 modeling methods for business processes can be found in Benedicenti et al. (1998) and Cox (1999; 2002). An object-oriented approach based on the Fuzzy-Set-Theory for the simulation of business processes is presented by Völkner, Werners (Völkner, 1998; Völkner & Werners, 2002).

Among others, Petri nets are used for the description of the dynamic aspects information systems. The bivalent behavior 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 behavior with fuzzy process conditions or incomplete, vague information, Petri-Nets were extended by fuzzy-concepts. The Fuzzy Petri net (Lipp, 1982) is created by the projection of several crisp Petri nets, in which the structural information is mapped as fuzzy sets.

Becker, Rehfeldt, Turowski (Becker, Rehfeldt, & Turowski, 1996; Rehfeldt, 1998) demonstrate the consideration of fuzzy data in business process modeling with the event-driven process chain on the example of industrial order processing. Vague sales information is seen as important, fuzzy exogenous input data, which is then 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 (Thomas, Hüsselmann, & Adam, 2002; Adam & Thomas, 2005; Adam, Thomas, & Loos, 2006) examine, 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 idea developed by the authors was extended and formalized in this article.

SUMMARY AND FUTURE RESEARCH

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 as a branch of soft computing. Second, it was carried out on the example of an established modeling language for business processes, the Event-driven Process Chain. The concept corresponds in a figurative sense with a "level extension" of the description 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 modeling 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 behavior can be understood relatively easily and existing knowledge can be integrated simply. Old rules can be taken over directly or marginally modified for the modification of the process to be controlled. This makes the constant improvement of the process definitions in the sense of a continuous process improvement easier.

The authors see a future challenges 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 to improve business processes through learning is currently being discussed (Adam, Thomas, & Loos, 2006).

REFERENCES

1. Adam, O. & Thomas, O. (2005). A Fuzzy Based Approach to the Improvement of Business Processes. In M. Castellanos & T. Weijters (Eds.), *BPI'05 : Workshop on Business Process Intelligence ; Nancy, France, September 5, 2005* (pp. 25–35). Nancy.
2. Adam, O., Thomas, O., & Loos, P. (2006). Soft Business Process Intelligence – Verbesserung von Geschäftsprozessen mit Neuro-Fuzzy-Methoden. In F. Lehner, H. Nösekabel, & P. Kleinschmidt (Eds.), *Multikonferenz Wirtschaftsinformatik 2006 : Band 2* (pp. 57–69). Berlin: GITO. (in German)
3. Becker, J., Kugeler, M., & Rosemann, M. (Eds.) (2005). *Prozessmanagement*. (5th ed.) Berlin: Springer. (in German)
4. Becker, J., Rehfeldt, M., & Turowski, K. (1996). Auftragsabwicklung mit unscharfen Daten in der Industrie. In J. Biethahn, A. Hönerloh, J. Kuhl, & V. Nissen (Eds.), *Betriebliche Anwendungen von Fuzzy-Technologien : Tagungsband zum 2. Göttinger Symposium Softcomputing am 29. Feb. 1996 an der Universität Göttingen* (pp. 51–61). Göttingen: Univ. Göttingen, Inst. für Wirtschaftsinformatik, Abt. I. (in German)
5. Benedicenti, L., Succi, G., Vernazza, T., & Valerio, A. (1998). Object Oriented Process Modeling with Fuzzy Logic. In J. Carroll, G. B. Lamong, D. Oppenheim, K. M. George, & B. Bryant (Eds.), *Applied computing 1998 : Proceedings of the 1998 ACM Symposium on Applied Computing ; Atlanta Marriott Marquis Hotel, Atlanta, Georgia, February 27-March 1, 1998* (pp. 267–271). Danvers, MA: ACM Press.
6. Blechar, M. J. & Sinur, J. (2006). *Magic Quadrant for Business Process Analysis Tools, 2006*. Stamford, CT: Gartner Research.
7. Borgelt, C., Kruse, R., Nauck, D., & Klawonn, F. (2003). *Neuro-Fuzzy-Systeme : Von den Grundlagen künstlicher Neuronaler Netze zur Kopplung mit Fuzzy-Systemen*. (3rd ed.) Wiesbaden: Vieweg. (in German)
8. Cox, E. (1999). Striving for Imprecision: Fuzzy Knowledge Bases for Business Process Modeling. *PC AI, 13*(4).
9. Cox, E. (2002). Knowledge-Based Business Process Modeling: Complex Systems Design Through A Fusion of Computational Intelligence And Object-Oriented Models. *PC AI, 16*(2), 15–23.
10. Dehnert, J. & Rittgen, P. (2001). Relaxed Soundness of Business Processes. In K. R. Dittrich, A. Geppert, & M. C. Norrie (Eds.), *Proc. of the 13th Conference on Advanced Information Systems Engineering (CAISE'01), Interlaken, Switzerland* (pp. 157–170). Berlin: Springer.
11. Dumas, M., van der Aalst, W. M. P., & ter Hofstede, A. H. M. (Eds.) (2005). *Process-aware Information Systems : Bridging People and Software through Process Technology*. Hoboken, New Jersey: Wiley.
12. Forte, M. (2002). *Unschärfen in Geschäftsprozessen*. Berlin: Weißensee. (in German)

13. Hoffmann, W., Kirsch, J., & Scheer, A.-W. (1992). *Modellierung mit Ereignisgesteuerten Prozeßketten : Methodenhandbuch ; Stand: Dezember 1992* (Rep. No. 101). Saarbrücken: Universität des Saarlandes. (in German)
14. Hüsselmann, C. (2003). *Fuzzy-Geschäftsprozessmanagement*. Lohmar: Eul. (in German)
15. Keller, G., Nüttgens, M., & Scheer, A.-W. (1992). *Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)"* (Rep. No. 89). Saarbrücken: Universität des Saarlandes. (in German)
16. Keller, G. & Teufel, T. (1999). *SAP R/3 prozeßorientiert anwenden : Iteratives Prozess-Prototyping mit Ereignisgesteuerten Prozessketten und Knowledge Maps*. (3rd ed.) Bonn: Addison-Wesley. (in German)
17. Kindler, E. (2004). On the Semantics of EPCs : A Framework for Resolving the Vicious Circle. In J. Desel, B. Pernici, & M. Weske (Eds.), *Business Process Management : Second International Conference (BPM 2004), Potsdam, Germany, June 17–18, 2004 ; Proceedings* (pp. 82–97). Springer.
18. Kindler, E. (2006). On the semantics of EPCs: Resolving the vicious circle. *Data & Knowledge Engineering, 56*(1), 23–40.
19. Langner, P., Schneider, C., & Wehler, J. (1998). Petri Net Based Certification of Event driven Process Chains. In J. Desel & M. Silva (Eds.), *Application and theory of Petri nets 1998 : 19th international conference ; proceedings* (pp. 286–305). Berlin: Springer.
20. Lipp, H.-P. (1982). Anwendung eines Fuzzy Petri Netzes zur Beschreibung von Koordinationssteuerungen in komplexen Produktionssystemen. *Wissenschaftliche Zeitschrift der Technischen Universität Karl-Marx-Stadt, 24*(5), 633–639. (in German)
21. Nüttgens, M. & Rump, F. J. (2002). Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In J. Desel & M. Weske (Eds.), *Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen (Promise '2002) : Gemeinsames Fachgruppentreffen der GI-Fachgruppen "Petrinetze und verwandte Systemmodelle" und "Entwicklungsmethoden für Informationssysteme und ihre Anwendung" (EMISA), Hasso-Plattner-Institut für Softwaresystemtechnik an der Universität Potsdam, 9.–11. Oktober 2002* (pp. 64–77). Bonn: Köllen. (in German)
22. Österle, H. & Winter, R. (2003). Business Engineering. In H. Österle & R. Winter (Eds.), *Business Engineering : Auf dem Weg zum Unternehmen des Informationszeitalters* (2nd ed., pp. 3–19). Berlin: Springer. (in German)
23. Rehfeldt, M. (1998). *Koordination der Auftragsabwicklung : Verwendung von unscharfen Informationen*. Wiesbaden: DUV. (in German)
24. Rittgen, P. (2000). Quo vadis EPK in ARIS? – Ansätze zu syntaktischen Erweiterungen und einer formalen Semantik. *Wirtschaftsinformatik, 42*(1), 27–35. (in German)
25. Rosemann, M. & van der Aalst, W. M. P. (2006). A configurable reference modelling language. *Information Systems*. in Press
26. Scheer, A.-W. (1997). *Business Process Engineering : Reference Models for Industrial Enterprises*. (7th ed.) Berlin: Springer.
27. Scheer, A.-W. (2001). *ARIS – Business Process Frameworks*. (4th ed.) Berlin: Springer.
28. Scheer, A.-W. (2002). *ARIS – Business Process Modeling.* (4th ed.) Berlin: Springer.
29. Scheer, A.-W., Thomas, O., & Adam, O. (2005). Process Modeling Using Event-driven Process Chains. In M. Dumas, W. M. P. van der Aalst, & A. H. M. ter Hofstede (Eds.), *Process-aware Information Systems : Bridging People and Software through Process Technology* (pp. 119–145). Hoboken, New Jersey: Wiley.
30. Thomas, O. (2006). Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation. In C. Bussler & A. Haller (Eds.), *Business Process Management Workshops : BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM, WSCOBPM, BPS, Nancy, France, September 5, 2005 ; Revised Selected Papers* (pp. 484–496). Berlin: Springer.
31. Thomas, O., Hüsselmann, C., & Adam, O. (2002). Fuzzy-Ereignisgesteuerte Prozessketten : Geschäftsprozessmodellierung unter Berücksichtigung unscharfer Daten. In M. Nüttgens & F. J. Rump (Eds.), *EPK 2002 : Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten ; Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)", 21.–22. November in Trier, Proceedings* (pp. 7–16). Bonn: GI. (in German)
32. Tietze, M. (1999). *Einsatzmöglichkeiten der Fuzzy-Set-Theorie zur Modellierung von Unschärfe in Unternehmensplanspielen*. Göttingen: Unitext. (in German)

33. van der Aalst, W. M. P. (1999). Formalization and verification of event-driven process chains. *Information and Software Technology, 41*(10), 639–650.
34. van der Aalst, W. M. P., Desel, J., & Kindler, E. (2002). On the semantics of EPCs: A vicious circle. In M. Nüttgens & F. J. Rump (Eds.), *EPK 2002 : Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten ; Workshop der Gesellschaft für Informatik e.V. (GI) und Treffen ihres Arbeitskreises "Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)", 21.–22. November in Trier, Proceedings* (pp. 71–79). Bonn: GI.
35. Völkner, P. (1998). *Modellbasierte Planung von Geschäftsprozeßabläufen : Entwicklung eines Entscheidungsunterstützungssystems auf Grundlage objektorientierter Simulation*. Wiesbaden: Gabler. (in German)
36. Völkner, P. & Werners, B. (2002). A simulation-based decision support system for business process planning. *Fuzzy Sets and Systems, 125*(3), 275–288.
37. vom Brocke, J. (2003). *Referenzmodellierung : Gestaltung und Verteilung von Konstruktionsprozessen*. Berlin: Logos. (in German)
38. Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338–353.
39. Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics, 3*(1), 24–44.
40. Zimmermann, H.-J., Angenstenberger, J., Lieven, K., & Weber, R. (Eds.) (1993). *Fuzzy-Technologien : Prinzipien, Werkzeuge, Potentiale*. Düsseldorf: VDI-Verl. (in German)
41. Zvieli, A. & Chen, P. P.-S. (1986). Entity-Relationship Modeling and Fuzzy Databases. In *International Conference on Data Engineering, February 5–7, 1986, Bonaventure Hotel, Los Angeles, California, USA* (pp. 320–327). Washington, DC: IEEE Computer Society Press.