Towards the Interchange of Fuzzy-EPCs: An XML-based Approach for Fuzzy Business Process Engineering

Oliver Thomas, Thorsten Dollmann

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

Abstract: Recent research has led to proposals for the integration of fuzzy based information and decision rules in business process models with use of concepts based on the fuzzy set theory. While the proposed Fuzzy EPCs provide an adequate method for conceptual representation of fuzzy business process models, the issue of exchanging and transforming such models, together with its enclosed executable components, with other dedicated information systems such as workflow management systems and fuzzy modeling tools has not been approached yet. As a first step in this direction, this paper proposes a machine-readable Fuzzy-EPC representation in XML based on the EPC Markup Language (EPML). We start with the formal Fuzzy-EPC syntax definition and introduce our extensions to EPML. At the same time, we highlight future application areas of the Fuzzy-EPC schema.

1 Introduction

Many concepts have been developed for designing and improving business processes, for generalizing them in reference models and for their use in implementation projects. A large number of these approaches emphasize the intuitive usability of methods by approximating them with human ways of thinking. However, one usually requires the exact quantification of the decision rules in necessary decision-making situations. Often however, for business processes, only uncertain, imprecise and vague information is available on the often not technically determined procedures. In addition, the underlying target system is usually characterized by unclear formulations and implicit interdependencies. This is for example illustrated by the statement “the processing time for commissions with *very high* priority should be reduced *considerably* while retaining a *high* processing quality by way of reducing the processing intensity *proportionately*”. In this example, neither the concrete parameter values of both the named goals with regard to processing time and processing quality, nor the derived measures can be quantified and thus, made directly processable without loss of information. Fuzziness is however, not only found in the meaning of the linguistic terms used when modeling expert knowledge. The knowledge formulated in this manner and the decision-making process, which reverts back to this knowledge also contain fuzziness.

The fuzzy event-driven process chain has been in discussion for over a decade as a means of considering this form of fuzziness. For example, Becker, Rehfeldt, Turowski [5; 14; 15] exemplarily demonstrate the consideration of fuzzy data in process modeling with event-driven process chains on the example of industrial order processing. Vague sales information transformed into tentative customer commissions is studied as relevant, exogenous input data with fuzziness in the form of uncertainty. Forte [7] also strives to extend the event-driven process chain in order to illustrate fuzziness. There he takes data and information objects into account. His extension is oriented on keeping the structure and the logic of the EPC itself and in addition, being able to derive new notations systematically and use them easily. A further approach to a fuzzy extension of the EPC is taken by Hüsselmann [8]. At first, he sees no necessity to extend the EPC syntactically for the inclusion of fuzzy data in business processes. He however also, introduces methods for illustrating fuzziness in EPC models using new constructs on the basis of the extension of Petri-nets to fuzzy Petri-nets [10]: fuzzy functions, fuzzy events, fuzzy operators and the mapping of fuzzy resource capacities. Thomas and Adam [3; 4] and other co-authors investigate how fuzzy data and its implementation in application systems can be used for the design of knowledge-intensive and weakly structured business processes. They also use a fuzzy extended EPC.

The existing works on fuzzy EPCs have two essential similarities: *First*, they deal with modeling 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 modeling tools and on the other, fuzzy systems. In doing so, the modeling-technical integration must be supported by a suitable information technical design. Still, despite the many studies available, up to now no interchange and storage format exists for the Fuzzy EPC that is supported by both classes of tools. One desirable requirement for the design of an interchange and storage format for fuzzy business process models is that the models and model changes created can be transferred and stored in standardized form, platform-independent as XML documents. The specification of Fuzzy EPC models with an XML schema connected with this is the topic of our paper. EPML [13] forms the foundation for this specification.

After the introduction of the Fuzzy EPC and its formal principles and syntactic elements in Chapter 2, the requirements to the XML-based specification of the Fuzzy-EPC will be described with EPML in Chapter 3. The interchange format Fuzzy-EPML will be derived in Chapter 4 and parts of it will be specified. Related work will be discussed in Chapter 5. The article will close in Chapter 6 with a summary and an outlook on future research challenges.

2 Background: The Fuzzy-EPC

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 directed 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. Further statements result through the use of the EPC as a central modeling language for the architecture of integrated information systems (ARIS) [16; 17]. These are based on the ARIS view concept. They are made through the annotation of other language constructs on EPC functions [18]. 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. The linkage of constructs, which 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.

In this article, we chose the EPC elements of the organization, data and output view as additional artifacts for process modeling, 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 (for further details cp. [19]).

We define a Fuzzy-EPC-model $FEPC = (E, F, C, A, O, D, L, R, M, FC)$ as an ARIS EPC model enriched with the following properties [19]:

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 between control flow objects and the various artifacts. 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.

Fig. 1 shows an example of the reference process for customer order processing. The process is represented in the form of a Fuzzy EPC. The fuzzy constructs of the EPC are characterized by grey shading.

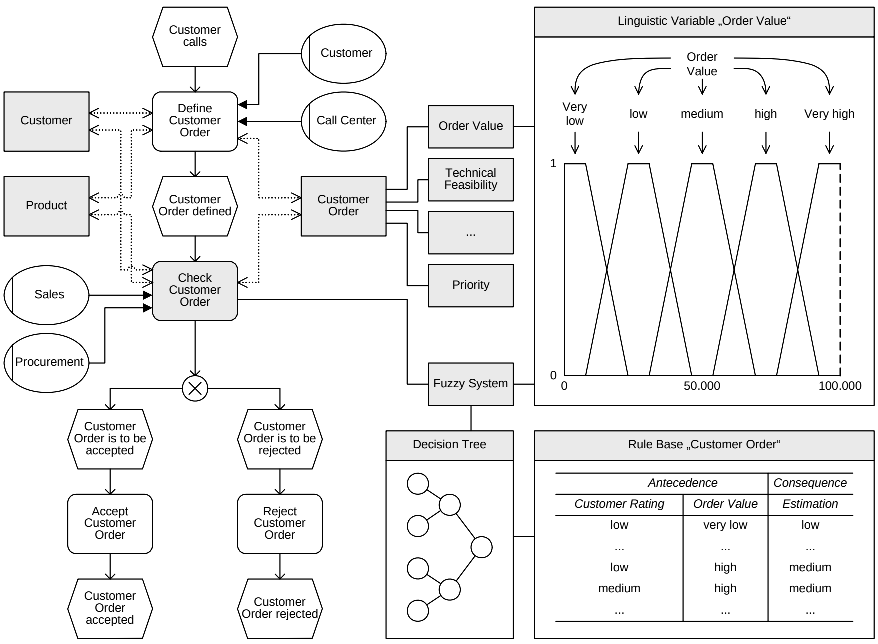

Fig. 1. Example of a Fuzzy EPC model

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 value of the order and customer rating. 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. In Fig. 1 the object attribute “Order Value” of the data object type “Customer order” is shown. It has the linguistic variables “very low”, “low”, “medium”, “high” and “very high” as terms.

4 XML-Schema for Fuzzy-EPCs

In this article EPML [13] will be systematically extended by structures for considering fuzzy information and thus, be developed as a representational format for the fuzzy event-driven process chain. In analogy to terminologies used up to now, the extension of the EPML will be referred to as *Fuzzy-EPML*. In the following, the most important modifications on the XML schema of the EPML will be discussed and the respective alteration constructions justified. The extensions will be shown in part only due to reasons of space.

Forthcoming from our preceding elaborations, we must seek a specification that allows for a representation of Fuzzy-EPCs in accordance with the definitions specified in the previous section. New EPML elements need to be defined. These elements have to capture the Fuzzy-EPC extensions, i.e. fuzzy functions, fuzzy attributes und fuzzy systems together with its components such as rules, linguistic variables, membership functions and parameters that induce the inference while process execution.

The extension of the EPML with fuzzy concepts must be carried out so that syntactically correct documents from the old schema version also remain valid concerning the new schema. This can be done by embedding new model elements into the schema of the existing language constructs. Thus, a new EPML element for fuzzy functions is not introduced, but rather the fuzzy information is added as an optional element of the function in EPML. In addition, the fuzzy attributes from ARIS language artifacts were introduced so that they can optionally carry the desired fuzzy values.

One essential extension of the EPML can be found in the XML schema definition of fuzzy functions. The extension is made by assigning a maximum of one `ruleSystem`-element to the corresponding process function on the basis of the `choice`-construct. This element is based on the type `typeRuleSystem`, which is specialized technically and syntactically through the types “fuzzy system” and “control flow fuzzy classificatory”.

```
<xs:complexType name="typeFunction">
   <xs:sequence>
      …
      <xs:choice minOccurs="0" maxOccurs="1">
         <xs:element name="ruleSystem"
          type="typeRuleSystem"/ >
```

```
      </ xs:choice>
   </ xs:sequence>
   …
</ xs:complexType>
```

Fuzzy-systems for decision support resp. authorized process execution should be used at XOR operators in EPC models. The respective extensions must now be supplemented for the derivation of the previously abstractly defined basis type `typeRuleSystem`. The control flow fuzzy classification system is characterized in particular by a special type of rule base. This rule is characterized on the upper level by the two elements `antecedent` and `consequence`. The first element maps the conjunctively linked part of the rules with the same name and has one or more `fuzzyProposition`-elements. The corresponding type contains an atomic fuzzy statement, which is made up of the elements `linguisticVariableId` and `termVariableId` in the schema. The statement "total order value is high" is thus expressed at this position by the reference of a linguistic variable and one of its terms. The `consequence`-part of the rule consists in this case of the reference of an event. This information is stored in the attribute `decisionToEvent`.

```
<xs:complexType name="typeFlowFuzzyClassificationSystem">
   <xs:complexContent>
      <xs:extension base="typeRuleSystem">
         <xs:sequence>
            <xs:element name="ruleBase"
             type="typeControlFlowFuzzyRuleBase"/ >
         </ xs:sequence>
      </ xs:extension>
   </ xs:complexContent>
</ xs:complexType>
```

A further main component of every fuzzy rule system is the basis of the underlying variables, with which the fuzzy information can be explicated in Fuzzy-EPC models. Such a basis is a series of linguistic variables, whose schema specification is structured in the following manner. Characteristic for the type `typeLinguisticVariable` – beyond the obligatory `id`-, `defRef`- and `comment`-attributes for the central language elements in EPML – is in particular, the attribute `label`, which carries the name of the linguistic variables and should correspond with the attribute to which a direct relationship is made by way of the reference with the entry `attributeIdRef`. This relationship expresses the connection from the process world to the "fuzzy world". With the attribute `typeRef` the variable is linked with the type of the corresponding attribute, which is defined in EPML and in this manner, inserted into the collection of attribute types in EPML. The content information for a linguistic variable can, in addition, be found in the elements `domain` (value domain), as well as in the listing of one or more `linguisticTerm`-elements (linguistic terms).

A linguistic term is characterized by an attribute `label`, as well as by an element `membershipFunction`. While the first carries the linguistic information in the form of the corresponding identifier, the semantics of this identifier are found in the second element. The element's type is specified by the type `typeMembershipFunction`. In the schema, special types of membership functions are built in. These cannot however, be discussed due to the length of the article. Each type is characterized by its own parameters, which were, once again, defined to meet the claim of "readability" against the background of data exchange and for the interpretation by software tools.

```
<xs:complexType name="typeLinguisticVariable">
   <xs:sequence>
      <xs:element name="linguisticTerm"
       type="typeLinguisticTerm" minOccurs="1"
       maxOccurs="unbounded"/>
      <xs:element name="domain" type="typeDomain"
       minOccurs="1" maxOccurs="1"/>
   </xs:sequence>
   <xs:attribute name="label" type="xs:string"
    use="required"/>
   …
</xs:complexType>
```

A basic concept for the definition of the Fuzzy-EPC was the extension of the EPC by way of attributes. The object types in EPC models, for example the individual data and output objects, as well as the organizational units have attributes, which are used to describe the individual object instances, as well as for their representation in application systems. Each attribute has a value domain, which defines the set of possible attribute values. For example, the value domain of the attribute "total order value" of a data object type "sales order" can be defined as a set of natural numbers. At the same time, the value set for the attribute "name" of the object type "customer" can be defined to the set of character strings, which consists of alphabetical symbols.

The respective attribute values, for example the amount of turnover in Euro [€] or the customer rating in percent [%], represent the relevant information on the decision node and therefore, require special consideration in the Fuzzy-EPC approach.

In order however, to use the special attributes and their values required for the representation of Fuzzy EPC-models, special attribute types must be defined. This is done by first introducing a new, abstract attribute type `typeExtendedAttribute`, which is derived from the EPML basic type `typeAttribute`. Based upon this type, new specific attribute types with individual `value` fields are derived to cope with special feature values. Among these qualitatively described characteristics based on the data type `string`, crisp numerical values on the basis of the data type `double` and fuzzily described feature presentations based on a membership function are defined. Thus, fuzzy attribute values, i.e. feature presentations in the form of a membership function, are describable. In addition, this fuzzy attribute can be given a qualitative description by way of the XML attribute `label`.

```
<xs:complexType name="typeFuzzyAttribute">
   <xs:complexContent>
      <xs:extension base="typeExtendedAttribute">
         <xs:sequence>
            <xs:element name="value"
             type="epml:typeMembershipFunction"
             minOccurs="0" maxOccurs="1"/>
```

```
        </ xs:sequence>
        <xs:attribute name="qualitativeLabel" use="optional"
     </ xs:extension>
  </ xs:complexContent>
</ xs:complexType>
```

Based on the said extension, data and performance objects, as well as organizational units can be extended by the additional attribute types by referencing the respective type in instance documents.

5 Application Scenario

In the following, the use of the Fuzzy-EPC markup language will be demonstrated in an application scenario. The listing shows an excerpt from the Fuzzy-EPML representation of the demo example for sales order checks. The focus is on the specification of the fuzzy control flow classification system, with which decisions about the acceptance or rejection of orders are made.

```
<epml>
…
<epc epcId="1" name="Scenario check customer order">
   …
   <function id ="6">
      <name>Check customer order</ name>
      <toProcess linkToEpcId="2"/ >
   </ function>
   …
</ epc>
<epc epcId="2" name="Check customer order">
…
<function id="23">
   <name>Decide on order acceptance or order rejection</ name>
   …
   <ruleSystem
    xsi:type="epml:typeFlowFuzzyClassificationSystem" id="24">
      <defuzzMethod>linear</ defuzzMethod>
      <fuzzyOperators>
         <accumulationOperator>
            <parameterizedOperator name="compensatoryAnd">
               <parameter name="gamma" value="0.5"/ >
            </ parameterizedOperator>
         </ accumulationOperator>
         <aggregationOperator>
            <standardOperator name="maximum"/ >
         </ aggregationOperator>
         <implicationOperator>
            <standardOperator name="minimum"/ >
         </ implicationOperator>
      </ fuzzyOperators>
      <variableBase>
         <linguisticVariable label="Estimation" id="300"
          attributeIdRef="40">
      …
      </ variableBase>
```

```
      <ruleBase id="361">
         <rule id="362">
            <antecedent>
               <fuzzyProposition>
                  <linguisticVariableId>300</ linguisticVariableId>
                  <linguisticTermId>301</ linguisticTermId>
               </ fuzzyProposition>
            </ antecedent>
            <consequence decisionToEvent="58"/ >
         </ rule>
      </ ruleBase>
      <eventShortlist idRefs="58 59"/ >
   </ ruleSystem>
</ function>
…
   <event id="58" defRef="008">
      <name>Customer order is to be accepted</ name>
      …
   </ event>
   <event id="59" defRef="009">
      <name> Customer order is to be rejected</ name>
      …
   </ event>
   …
</ epc>
</ epml>
```

It is apparent that the scenario "sales order check" from Fig. 1 is stored as an EPC model with the `epcId="1"` in EPML. The function Check sales order is organized hierarchically and linked with a second EPC model in EPML code, which maps the decision hierarchy concerning the acceptance or rejection of the sales order. In the listing, the specification of the fuzzy function Decide on order acceptance or rejection is highlighted from this EPC model. With it, the said decision is made. The corresponding control flow fuzzy classification system is stored in Fuzzy-EPML in the element `ruleSystem`. In order for the model to serve as a basis for the inference during runtime, different fuzzy parameters must be selected. In the previous listing it is clear that the parameter selection in Fuzzy-EPML is stored by stating the corresponding parameters. Linear defuzzification was for example, selected as a defuzzification method and even the selection of the required operators for the evaluation/ analysis of the rules is documented in the corresponding component `fuzzyOperators`. The Fuzzy-EPC model is completely represented in Fuzzy-EPML with the described components.

5 Related Work

There are some contributions in current literature, which introduce interchange formats into the field of soft computing on the basis of XML. Thus for example, LEE, FANJIANG [9] develop a specific XML schema, with which fuzzy information on fuzzy-theoretic concepts can be taken over in an object-oriented data model. This approach does not however, provide a schema on the basis of which a comprehensive specification of fuzzy systems with modules for inference can be supported beyond the specification of fuzzy attributes.

A similar focus can be seen by WITTE [22], who introduces new data types for a fuzzy XML format with the help of a document type definition for the data exchange between different fuzzy information systems. To do so, object-oriented fuzzy models are discussed. Thus, for WITTE fuzzification approaches are also the focus of his studies, which – beyond an attribution – focus class hierarchies and inheritances resp. gradual degrees of membership to a class.

Central questions on the fuzzification of conceptual data modeling are also the focus for MA [11; 12]. Thus, from the view of fuzzy process modeling few of the findings can be transferred to a domain-specific XML schema. While proprietary schematizations of fuzzy attributes are introduced in part, the focus remains on the, seen from the view of process modeling, subtle data structure relationships.

DE SOTO, CAPDEVILA, FERNÁNDEZ [6] provide a first dedicated schema design for the representation of fuzzy-systems in XML with iXSC (Extensible Soft Computing Language). Their approach follows the principle of reuse. Analogue to the logical structure of EPML, the composition of complex constructs takes place here on the basis of simple elements with the help of type definitions.

TUROWSKI, WENG [2] also meet the claim of integrating fuzzy application systems on different platforms and common application systems with the definition of the XML data types of a fuzzy rule base in the form of a document type definition. Their approach contains no cross-references in the schema or a similar linking concept, so that redundant information (in particular, linguistic variables and membership functions) cannot be avoided in the sub-tree of a rule base.

The ABLE Rule Language (ARL) from the IBM Corporation [1] forms a further XML schema definition for fuzzy rule bases. Here a sub-set of the language for the specification of fuzzy-if-then-else rules, crisp and fuzzy input and output variables with membership functions of a special type are definable.

TSENG, KHAMISY, VU [20; 21] dedicate themselves explicitly to the problem of the universal representation of fuzzy systems in XML for the integration of different representation formats. Against the background of the growing number of product-dependent and thus, proprietary formats, the interchange between different application systems, as well as their extension by means of fuzzy concepts should thus be facilitated. It is a matter of several XSD files, which depending on requirements, can be imported into one another, without a referencing and identification mechanism.

In conclusion, it can be stated that many of the approaches discussed only provide proprietary language definitions for standard fuzzy systems. Some of the approaches cannot avoid a large amount of redundant information because they omit identification and referencing mechanisms and others go without specifying a standard vocabulary for operators and membership functions and thus, limit the universality of the language definition.

It should also be emphasized that none of the said approaches addresses the field of “fuzzy classification systems” explicitly. This is, by all means, justified from a fuzzytheoretical view through the embedding of classification systems as special fuzzy rule systems. However, this study is a "prime example" for the application of fuzzy classification systems with its special decision rules regarding the execution of business processes in a company and thus, justifies the independent treatment of such a fuzzy classification system for the planned extension of EPML. Furthermore, "normal" fuzzy rules should continue to play a role when multi-level decision hierarchies are to be illustrated with "multi-level" fuzzy systems.

6 Summary and Outlook

This paper presented an XML schema-based specification of the Fuzzy-EPC using the EPML format. We reported on the design of the Fuzzy-EPC compliant schema and showed major syntactical extensions. Furthermore we sketched a realistic example (sales order checks) showing that Fuzzy-EPML is able to serve as an adequate interchange and storage format for fuzzy business process models.

Since fuzzy business process models can now be transferred and stored in a standardized form and platform independent as XML documents, some further tasks will be approached in the near future. First, there is not yet adequate tool support for the modeling of Fuzzy EPCs. However, as the schema is now available, we are currently working on respective tool support. Furthermore the task of transforming Fuzzy-EPC models for execution or for further analysis will be approached. Here we emphasize the further processing of model parts in fuzzy modeling tools with process improvement in combination with learning algorithms. Since the lack of integration between fuzzy applications and common business application systems is softened, the integration of other soft computing techniques to automatically discover process model parts and decision rules looks promising and is currently evaluated.

References

[1] IBM (ed.): *ABLE Rule Language: User's Guide and Reference, Version 2.3.0.* IBM Corporation, 2005

[2] Ackermann, J. et al.: Vereinheitlichte Spezifikation von Fachkomponenten – Memorandum des Arbeitskreises 5.10.3 Komponentenorientierte betriebliche Anwendungssysteme, Februar 2002. Universität Augsburg, 2002

[3] 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.* Nancy, 2005, pp. 25–35

[4] Adam, O.; Thomas, O.; Loos, P.: Soft Business Process Intelligence – Verbesserung von Geschäftsprozessen mit Neuro-Fuzzy-Methoden. In: Lehner, F.; Nösekabel, H. ; Kleinschmidt, P. (eds.): *Multikonferenz Wirtschaftsinformatik 2006 : Band 2.* Berlin : GITO, 2006, pp. 57–69

[5] Becker, J.; Rehfeldt, M.; Turowski, K.; Vering, O.: A Fuzzy Approach to a Customer-Oriented Sales Workflow. In: Steele, N. C. (ed.): *ISFL '97 : Second International ICSC Symposium on Fuzzy Logic and Applications, February 12–14, 1997, Swiss Federal Institute of Technology Zurich, Switzerland.* Zürich : ICSC Academic Press, 1997, pp. 370–376

[6] de Soto, A. R.; Capdevila, C. A.; Fernández, E. C.: Fuzzy Systems and Neural Networks XML Schemas for Soft Computing . In: *Mathware and Soft Computing* 10 (2003), No. 2–3, pp. 43–56

[7] Forte, M.: *Unschärfen in Geschäftsprozessen.* Berlin : Weißensee, 2002

[8] Hüsselmann, C.: *Fuzzy-Geschäftsprozessmanagement.* Lohmar : Eul, 2003

[9] Lee, J.; Fanjiang, Y. Y.: Modeling imprecise requirements with XML. In: *Information and Software Technology* 45 (2003), No. 7, pp. 445–460

[10] Lipp, H. -P.: Anwendung eines Fuzzy Petri Netzes zur Beschreibung von Koordinationssteuerungen in komplexen Produktionssystemen. In: *Wissenschaftliche Zeitschrift der Technischen Universität Karl-Marx-Stadt* 24 (1982), No. 5, pp. 633–639

[11] Ma, Z.: *Fuzzy Database Modeling with XML.* Berlin : Springer, 2005

[12] Ma, Z.; Yan, L.: Fuzzy XML data modeling with the UML and relational data models. In: *Data & Knowledge Engineering* 63 (2007), No. 3, pp. 970–994

[13] Mendling, J.; Nüttgens, M.: EPC Markup Language (EPML) – An XML-Based Interchange Format for Event-Driven Process Chains (EPC). In: *Information Systems and e-Business Management* 4 (2006), No. 3, pp. 245–263

[14] Rehfeldt, M.: *Koordination der Auftragsabwicklung : Verwendung von unscharfen Informationen.* Wiesbaden : DUV, 1998

[15] 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, NM : TSI Press, 1996, pp. 585–589

[16] Scheer, A.-W.: *ARIS – business process frameworks.* 2. ed. Berlin : Springer, 1998

[17] Scheer, A.-W.: *ARIS – business process modeling.* 2. ed. Berlin : Springer, 1999

[18] 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, New Jersey : Wiley, 2005, pp. 119–145

[19] Thomas, O.; Dollmann, T.; Loos, P.: Towards Enhanced Business Process Models Based on Fuzzy Attributes and Rules. In: *Proceedings of the 13th Americas Conference on Information Systems : August 09–12, Keystone, Colorado, USA.* Atlanta, Georgia, USA : AIS, 2007

[20] Tseng, C.; Khamisy, W.: *XML Schema for Fuzzy Systems*. URL http://mh213d.cs.sjsu.edu/webintelligence/fuzzyschema/Schema.html#Schema [03.03.2006]

[21] Tseng, C.; Khamisy, W.; Vu, T.: Universal fuzzy system representation with XML. In: *Computer Standards & Interfaces* 28 (2005), No. 2, pp. 218–230

[22] Witte, R.: *Architektur von Fuzzy-Informationssystemen.* Norderstedt : Books on Demand, 2002