# Fuzzy Extended BPMN for Modelling Crime Analysis Processes

Paul Cotofrei and Kilian Stoffel

Information Management Institute,
University of Neuchâtel, Switzerland

**Abstract.** In the frame of an overall project concerning the development of an intelligent process-driven framework for crime analysis, the modelling phase of crime analysis processes requires formal approaches being able to capture both the vague nature of forensic data and the uncertainties and conjectures characterizing the inference structures of this domain. In this context, a first analysis on the feasibility of a fuzzy embedded BPMN using the extensibility mechanism introduced by BPMN 2.0 specification is considered[1].

## 1 Introduction

In recent years, mathematical, statistical and computational science methods have found extensive applications in developing new procedures for crime investigation, prosecution and the law enforcement. An important and promising research objective in Computational Forensic [1], an emerging interdisciplinary research domain, is the development of an intelligent framework in which sophisticated computational methods, driven by forensic processes, analyze data and extract potentially useful information.

The development of such a framework is based on the new paradigm of *domain driven data mining* [2], in which ubiquitous intelligence is incorporated into the mining processes and models, and a corresponding problem-solving system is designed for knowledge discovery and delivery. For the forensic domain, crime analysis processes can be considered as a representation of this ubiquitous intelligence (domain knowledge). Therefore, formal modelling of various forensic processes/procedures followed during crime analysis is a necessary phase in the design of any process-driven computational framework.

The formalism chosen for the realization of this task must satisfy two criteria: it has to be intuitive enough to be used by users without a strong computational science background (here, forensic experts), and rich enough to express all kinds of processes which may occur in crime analysis. Moreover, the representation of a process must be independent of the computational system/service executing the process, but must also be easily integrated in a process-driven data analysis system. One of the best-adapted solution is given by the approach proposed by the Business Process Model and Notation. The adaptability of this modelling

[1] This work was supported by SNF grant CR2012-135236/1

framework to correctly represent procedures, decisions, events, roles and organizational structures specific to the crime analysis domain is indirectly proven by a case study presented in [3], showing the suitability of BPMN to represent administrative processes inside an academic organization (institutions which, like police crime analysis units, have no business purposes). Moreover, the administrative processes in any academic unit are represented in written forms (legal texts) with semantic constraints, context very similar to the crime analysis process, which are running under a complex set of legal and administrative constraints.

But any methodology for modelling the inference structures that pertain to crime analysis must handle the specific nature of crime data (which represents also the main challenge in forensic science): the pieces of evidence are hidden in a mostly chaotic environment. Examples are a smudged fingermark on a glass, a fragmentary ear mark on a door, a disguised handwriting or an unobtrusive paint scratch. Moreover, traces found will never be identical to known specimen in a reference base, even if traces are caused by an identical source. As a consequence, decisions and deductions have to be performed on the basis of partial knowledge, approximations, uncertainties and conjectures [4]. The fuzziness of crime data is an intrinsic parameter of the knowledge inference process.

The possibilities for the integration of fuzziness into process modelling are limited for any of the existing methodologies/notations which require completeness and precision. This weakness derives from the difficulties of translating the vagueness of natural language descriptions or the environment's dynamic changes into new process models, and of adapting existing models. Therefore the transformation of real-world uncertainty, dynamics, and vagueness into crisp process models based on standard logic implies a loss of relevant information [5].

The imperfect nature of forensic data and the uncertainties and conjectures which characterize the inference structures in this domain may potentially conduct to the impossibility of BPMN to express certain crime analysis processes. Meanwhile, decisions based on vague or qualitative information, imprecise linguistic description of the environment, vagueness induced by ascertainment of reality based on scales, all these represent complex scenarios which may characterize also business processes. Therefore, an analysis about the feasibility of the extension of the modelling formalism towards a fuzzy-based BPMN (the overall goal of the paper) responds to a real issue in business process management too.

This paper is structured in the following way. After a short review of the papers related to the integration of fuzziness aspects in process modeling, we will introduce the basic notions and concepts of the theory of fuzzy sets (including fuzzy logic and fuzzy inference systems) used in this paper. The section 4 will describe the general crime analysis process and its related issues, with an emphasis on a particular process concerning the analysis of serial burglaries. The next section will introduce the core elements of the Business Process Model and Notation and will underline the major changes introduced by BPMN 2.0 current specification. The feasibility and the utility of a BPMN extension with fuzzy concepts will be analyzed in Sect. 6 through the answers given to some basic questions such as “what BPMN elements are worth to be extended with fuzzy concepts” or “what is the formal procedure for linking fuzzy attributes to an element of the BPMN 2.0 specification”. Finally, the paper will close with a summary and an outlook on future research challenges.

## 2 Literature review

Even if the concept of fuzzy sets was introduced by Lofti A. Zadeh more than forty ears ago [6], studies related to the integration of fuzziness aspects into information, resp. process modeling, are much more recent. Zviely and Chen [7] described fuzzy-extensions of Entity-Relationship-Models (ERM); the fuzzified data structures consequently leads to the processing of fuzzy data in the respective business processes. Fuzzy-based object-oriented modeling methods for business processes were proposed by Benedicenti et al. [8], Cox [9, 10] and Völkner & Werners [11]. Petri Nets, a formal mathematical modelling method, are widely used for modelling business processes. In order to represent system behaviors with fuzzy process conditions or incomplete, vague information, Petri-Nets were extended by fuzzy-concepts (Chen et al. [12], Looney [13] or Ye [14])). Fuzzy-extensions of the process modeling language EPC (Event-driven Process Chain) were extensively analyzed using the example of industrial order processing. Becker et al. [15] visualized fuzzy extensions of the process by shaded objects and demonstrated the usefulness of fuzzy data integration (vague sales information). Thomas and Adam [5, 16, 17] examined how fuzzy data can be used to design knowledge-intensive and weakly structured business processes and how to design an interchange and storage format for fuzzy business process models.

## 3 Fuzzy sets theory

The concept of fuzziness, as conceived in the business context, is related to the uncertainty of data and its interdependency in the human comprehension of the reality. Mathematically, fuzziness describes event ambiguity and is based on Fuzzy sets, introduced by Lotfi A. Zadeh [6]. They differ from the classical notion of sets by allowing the gradual assessment of the membership of elements. This is described with the aid of a membership functions valued in the real unit interval $[0, 1]$. Emerged from the development of the theory of fuzzy sets, fuzzy logic is an extension of the case of multi-valued logic, assigning to each proposition a degree of truth - a value varying between 1 (*absolutely true*) and 0 (*absolutely false*).

Fuzzy logic (together with neurocomputing and genetic algorithms) is one of the techniques of *soft computing*, i.e. computational methods tolerant to suboptimality, impreciseness (vagueness) and partial truth and giving quick, simple and sufficiently good solutions. The guiding principle of these methods is perfectly adapted to the way in which reasoning and deduction have to be performed in forensic science

### 3.1 Fuzzy Inference Systems

Describing generally vague concepts (as tall people, hot weather, morning hours, etc.), fuzzy sets associate a membership function (denoted $\mu(x)$) which maps an input value to its appropriate membership value. A membership function may be any arbitrary function with values in $[0, 1]$, but in practice basic functions are used, such as piece-wise linear functions, Gaussian distribution functions, the sigmoid curves, quadratic and cubic polynomial curves. In a mathematical notation, a fuzzy set is a pairs $A = \{(x, \mu(x)\}$. The set of elements that have a non-zero membership value is called the *support* of the fuzzy set.

Fuzzy sets are well suited to describe linguistic variables (taking as values words or sentences in a natural or artificial language [18]). As example, the *modus operandi* of a series of burglary cases includes the linguistic variable "burglary occurring time", which takes as values *morning, afternoon, evening* and *night.* Fig 1 displays the membership functions of each fuzzy set defined by these linguistic terms for input values representing daily hours. According to the membership definitions, a case occurred at 22.00 may be characterized as a "burglary during evening" with a value of 1 and as a "burglary during night" with a value of 0.4. *Remark*: there is a fuzziness aspect even in the definition of these fuzzy sets - it's obvious that *evening*, as example, has not the same support and membership function $\mu_{evening}$ if the burglary occurred during summer or during winter.

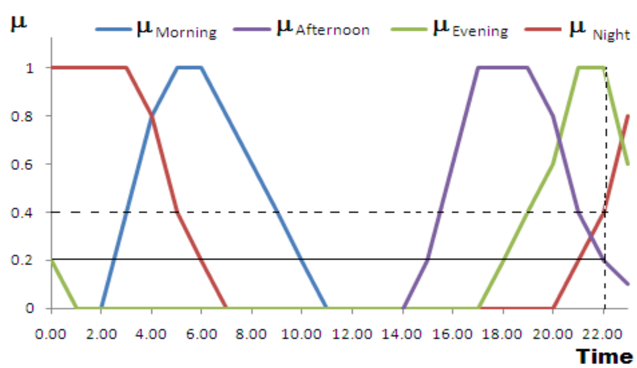


**Fig. 1.** Linguistic variable "burglary time".

The fuzzy logical reasoning is a superset of standard Boolean logic, i.e. the truth functions of connectives have to behave classically on the extremal values $0, 1$. For conjunction, a family of functions satisfying this condition is the set of binary T-norm operators [19] (*min* is a classical exemple), whereas for disjunction it's the set of binary T-conorm operators (*max* is a classical example). Several parameterized T-norms and dual T-conorms have been proposed in the literature, such as those of Yager[20], Dubois and Prade[21] and Sugeno [22].

A fuzzy rule *if-then* has the form *If $x$ is $A$ Then $y$ is $B$*, where $A$ and $B$ are fuzzy sets. Interpreting an *if-then* rule involves two distinct parts. Firstly, the premise of the rule is evaluated, which involves *fuzzifying* the input (i.e. calculate the membership value) and - if the premise has multiple parts - applying any necessary fuzzy operators. Secondly, the result is applied to the consequent (operation known as implication) using an *implication* function, which modifies the output fuzzy set to the degree specified by the antecedent. The modification is usually realized by truncation, using the *min* function, or by scaling, using the *prod* function, but other theoretical approaches have been proposed [23].

The fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The systems using fuzzy inference have been applied in different domains, such as automatic control, data classification, decision analysis, expert systems or computer vision. In the literature two types of fuzzy inference systems (FIS), differing in the way the output is determined, are the most known: Mamdani-type and Sugeno-type. The Mamdani's fuzzy inference method [24] expects the output membership functions to be fuzzy sets. For Sugeno-type systems [25], the output membership function is a singleton, which simplifies the defuzzification process. In general, Sugeno-type systems can be used to model any inference system in which the output membership functions are either linear or constant.

In the context of an universe comprising fuzzy sets and a number of weighted fuzzy rules *if-then*, a fuzzy inference process comprises five phases: (i) fuzzification of the input variables (those appearing in the premise part of the rules), (ii) application of the fuzzy operator (AND or OR) in the premise (if necessary), (iii) implication from the premise to the consequent, (iv) aggregation of the consequents across the rules, and (v) defuzzification.

The basic model for a fuzzy inference system considers that membership functions, representing the characteristics input, are predetermined by the user. In the situation where these characteristics can't be "guessed" only by looking at the data, a neuro-adaptive learning technique may be used to *learn* information about a data set, by choosing the parameters so as to tailor the membership functions to the input/output data. The final system is called an *adaptive neuro-fuzzy inference system* because it uses a network-type structure similar to that of a neural network for the learning purpose.

## 4 Crime analysis processes

Crime analysis can be defined as "the systematic study of crime and disorder problems as well as other police-related issues (including socio-demographic, spatial, and temporal factors) to assist the police in criminal apprehension, crime and disorder reduction, crime prevention, and evaluation" [26]. It is increasingly recognized that the analysis of crime should start from the physical marks collected at the scene. This elementary information on crime is measurable and directly comparable, while other types of information used in crime analysis, such as the *modus operandi* of the offender, are already uncertain reconstruction of past events [27, 28].

Crime analysis processes must be seen as a whole, from the collection of data to the practical use of analytical products in making decisions about how to respond to crime. Figure 2 shows an abstract description of such a process [29, 30].

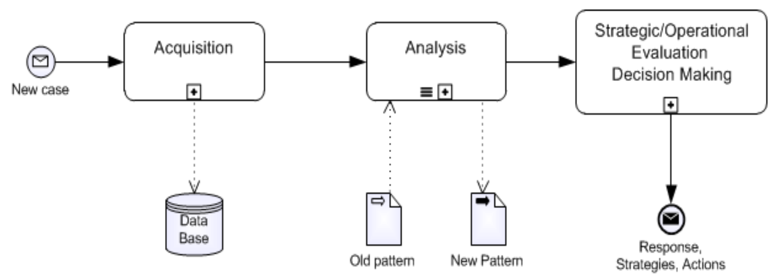


**Fig. 2.** Representation of a crime analysis process

The process can be described as the acquisition of data (e.g. through crime scene examination) into a structured memory that represent assumptions about the current criminal environment, the analysis of the information and the dissemination of intelligence that influences decision makers. In particular, the follow-up of high volume crimes and burglaries can be considered along these lines.

This representation clarifies many fundamental difficulties that have to be faced:

- The crime environment may rapidly evolve. Crime analysis process should be highly adaptable and executed in "real time".
- Marks collected from the scene are the fragmentary result of singular past events. Reconstruction of what occurred has an hypo-deductive logical nature and is based on imperfect data (approximate and non monotonic reasoning).
- Crime analysis processes are running under a complex set of constraints (legal, administrative, economic, time) that impose to make a trade off when designing a well balanced system.
- The process does not end once patterns are detected. These patterns must be interpreted and, according to the nature of the knowledge gained, feed strategic or tactical decision making processes. A system that generates irrelevant or "obvious" criminal patterns may thus be more of a handicap than a help in a practical environment.

The general crime analysis process is customized for specific crimes and specific type of decision. A large number of studies were conducted on burglary cases (break and enter combined with theft) and specially for the particular problem of serial burglary investigation [29, 31]. Serial crime analysis is an investigative process that is distinct from the process of evidence demonstration, even if both are linked. Thus, the interpretation and use of data collected is not for demonstration purposes and may even remain hidden or implicit for the court.

The most important data objects required/produced by activities included in the analyzing process of serial burglaries are [29]:

1. *case* - burglary instance, characterized by a number of attributs such as Modus Operandi, forensic science data, date and time, witness observations, stolen objects, etc..
2. *series* - cases perpetrated by the same author or group of offenders; the membership of a case to a series expresses a degree of certainty.
3. *profile* - synthetic description of a series, having as attribute geographic information, a physical description of the author(s), the type of shoes worn, the Modus Operandi, signatures, vehicles used, etc..

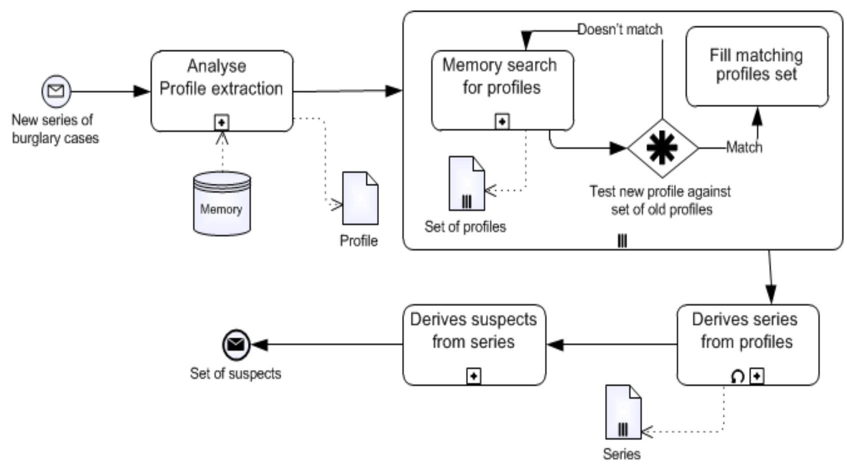


**Fig. 3.** Crime analysis process for a series of burglary

The process designed in Fig. 3 describes the situation where a new series of burglaries is detected. The first activity (sub-process) is the extraction of the corresponding profile. In a second sub-process, old profiles are tested against the new profile to detect similarities. If the degree of matching is high enough, the old profile is included in the set of matching profiles. The next activity derives the series corresponding to the matching profiles, which allows, during the last activity, the deduction of the set of convicted perpetrators representing the selection of suspects for the new series. Obviously, a very important problem "hidden" in the graphical description of the process is related to the fuzziness of the object attributes (i.e., time *moment of the day*) or logical tests (i.e., a case *belongs* to a series; a profile *matches* another profile).

## 5 Business Process Model and Notation

The domain knowledge, represented here as the collection of processes representing crime investigation activities, is modeled by forensic experts. The formalism chosen for the realization of this task must satisfy two criteria: intuitive enough to be used by users without a strong computational science background, rich enough to express all kinds of processes which may occur in crime analysis. Moreover, the representation of a process must be independent of the computational system/service executing the process, but must also be easily being integrated in a process-driven data analysis system. One of the best-adapted solution regarding our domain of interest is given by the Business Process Model and Notation approach.

Conceptually, BPMN is an agnostic methodology, i.e. a methodology which gives guidance as to the purpose and level of detail for modelling. BPMN provides businesses with the capability to understand and represent their internal business procedures with a graphical notation and gives organizations the ability to communicate these procedures in a standard manner. Furthermore, the graphical notation facilitates the understanding of high-performance collaborations and business transactions between the organizations. The modeling in BPMN is made by simple diagrams with a small set of graphical elements (see Fig. 4).

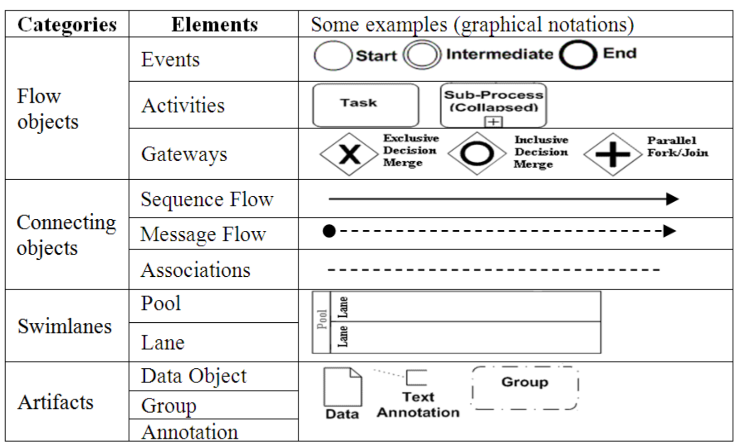


**Fig. 4.** BPMN diagram elements

Some of these graphical elements have been used in the flow-chart diagram of burglary analysis process (Fig. 3): Start Event (triggered by a message - new series of burglaries), Collapsed Sub-Processes (i.e. "Analyse/Profile extraction") or Expanded Sub-Process (with multiple parallel activities), Complex Gateway

("Test new profile.."), Data Object (single object "Profile" or collection of objects "Set of profiles"), Sequence Flows and End Event (generating a message).

Developed by Object Management Group (OMG), BPMN is conceived as a standard notation based on the best ideas from different - and divergent - notations and methodologies: UML Activity Diagrams, IDEF, ebXML BPSS, ADF Diagrams, RosettaNet, LOVeM, and Event-Process Chains (EPCs). The current version is BPMN 2.0, released in January 2011 [32]. The major changes, brought in order to eliminate known inconsistency and ambiguities of the previous version (BPMN 1.2) consist of (i) the formalization of the execution semantics for all BPMN elements, (ii) the definition of an extensibility mechanism for both Process model extensions and graphical extensions, (iii) the refinement of Event composition and correlation, (iv) the extension of the definition of human interactions, and (v) the definition of a choreography model.

An important remark is that, even if BPMN shows the flow of data (messages), and the association of data artifacts to activities, it is not a data flow language and does not support data and information models.

## 6 Fuzzy extension of BPMN

The diagram of the process for the analysis of a new burglary series (Fig. 3) expresses a verbal description of a complex reality (including activities, dynamic data, relationships between data and decisions, etc.). The vagueness and uncertainty of the real world is "hidden" in this model either by collapsing atomic tasks in generic sub-processes (as "Analyse/Profile extraction" or "Derive suspects from series") or by using vague, verbal description for gateway conditions (as "Profile match/don't match"). As long as the modeler intends to visualize the business process as a simple flow-chart diagram, the current specification of BPMN are completely sufficient. But if, in a second phase, the modeler intends to perform operational simulation, monitoring and deployment of the process (activities not supported by BPMN specification), the only approach is to map the appropriate visualization (a notation) to the appropriate execution format (a BPM execution language[2]). During the simulation/deployment phase (runtime), the vague and uncertain information must be integrated in the logic of the process execution (which implies that the execution language must be able to manipulate fuzzy logic). As a BPMN diagram may be mapped to more than one platform dependent process modeling language, the best policy (in our opinion) is to integrate the fuzzy aspects - if present - of reality (fuzzy data, fuzzy rules, fuzzy inference) in the early phase of process design, based on a fuzzy extended BPMN.

Embedding fuzziness in the modeling process phase rises three basic questions:

[2] BPMN provides the mapping of a sound (neither deadlocks nor lack of synchronization) orchestration process to the web service-based XML execution language WS-BPEL

1. What are the fuzzy logic concepts which may be represented?
2. What are the BPMN elements which may be extended with these concepts?
3. What is the mechanism allowing to link fuzzy information with BPMN elements?

For the first question, it is recommended that at least three concepts - fuzzy sets, fuzzy rules and fuzzy systems (in this order) - are to be represented. To define a fuzzy set, only the membership function ($\mu$) of the fuzzy set needs to be represented (usually, the domain may be derived from the membership representation). But the membership function representation is a difficult task due to potentially infinitely many types of such functions. The classical solution is to restrict these types to a limited number of basic functions (such as piece-wise linear, Gaussian distribution, sigmoid curves and cubic polynomial curves), each defined by a particular list of parameters. Concerning a fuzzy rule, its representation must contain (i) the *premise* part, (ii) the *consequence* part and (iii) the *implication* function. If we restrict ourselves to only conjunctive premises, then the representation of this element includes the function for the fuzzy operator *AND* and a collection of pairs $(x_i, \mu_i)$, where $x_i$ is an input variable and $\mu_i$ is a membership function. Similarly, the representation of the *consequence* part is the pair $(y, \mu)$, where $y$ is the output variable. Finally, a fuzzy system is represented as a set of fuzzy rules (in a set, all rules have the same output variable) accompanied by the function type for the aggregation operator and the defuzzification method type. Moreover, each fuzzy rule representation includes as supplementary information the rule's weight in the set.

Before to answer the second question, the concept of *fuzzy attribute* and those of *fuzzy constraint* must be introduced. Intuitively, the value domain of a fuzzy attribute consists of fuzzy sets. Furthermore (see [16]), such an attribute can be interpreted as a linguistic variable, which implies that the attribute name is the linguistic variable name and that the value domain of the attribute is, at the same time, the basic set of the linguistic variable. As example, "burglary time" is a fuzzy attribute taking as value one of the fuzzy sets *morning, afternoon, evening* and *night* (see Sect. 3). A fuzzy constraint is a binary function characterized by the assignment of a fuzzy system for which the defuzzification phase of the inference process returns a binary value (0 or 1), based on a cut-off parameter. The fuzzy sets from the *premise* or the *consequence* part of each rule are values of a defined fuzzy attribute. As example, being assigned a one-rule fuzzy system "IF `crime time` is *morning* AND `crime place` is *close to highway* THEN `matching score` is *average*", a function is a fuzzy constraint if it returns 1 for $\mu_{average}(y) \geq 0.7$ and 0 elsewhere (here, $y$ is the crisp output value and 0.7 the chosen cut-off parameter).

We call a BPMN element having attached at least a fuzzy attribute or a fuzzy constraint a *fuzzy element*. Among different BPMN modeling elements, those who naturally may be enriched with fuzzy attributes are Data Objects[3] (Data Input/Data Output) and Messages[4]. Due to the fact that some Events

[3] provide information about what Activities require to be performed and/or produced
[4] depict the contents of a communication between two Participants

(like the Message, Escalation, Error, Signal and Multiple Event) have the capability to carry data, these Event types becomes fuzzy Events if the associated Data Object is a fuzzy element. The BPMN elements which, by their meaning, may include a fuzzy constraint are the Gateways (Exclusive Gateway, Inclusive Gateway, Complex Gateway): in order to determine the process flow, one or more conditional expressions must be evaluated. There are also other particular elements which may evaluate conditional expressions - as Conditional Sequence Flow and Conditional Event - which give fuzzy Sequence Flows and fuzzy Events. An interesting case is an Event-base Gateway using an Intermediate Message Event for flow control: if the attached Message is a fuzzy object, then the Gateway itself becomes a fuzzy element through a fuzzy attribute.

### 6.1 Extension mechanism

The key permitting the extension of BPMN with fuzzy concepts is given by the introduction, in the current version of specification (2.0), of an extensibility mechanism that allows extending standard BPMN elements.

As described in [32], the technical structuring of BPMN is based on the concept of extensibility layers on top of a basic series of simple elements identified as Core Elements. From this core set of constructs, layering is used to describe additional elements of the specification that extend and add new constructs to the specification and relies on clear dependency paths for resolution. The XML Schema model lends itself particularly well to the structuring model with formalized import and resolution mechanics that remove ambiguities in the definitions of elements in the outer layers of the specification.

The extensibility mechanism provides a set of extension elements, which allows BPMN adopters to attach additional attributes and elements to standard and existing BPMN elements. This approach results in more interchangeable models, because syntactically correct diagrams from the standard schema version also remain valid concerning the extended schema. On the other hand, it is the responsibility of the modelers that, by adding non-standard elements or Artifacts to satisfy a specific need, the extension attributes does not contradict the semantics of any BPMN element.

The core elements of a BPMN Extension are the `ExtensionDefinition` and `ExtensionAttributeDefinition`. The first element can be created independent of any BPMN element. The second defines a list of attributes that can be attached to any BPMN element. The attribute list defines the name and type of the new attribute. Theoretically every BPMN element which subclasses the BPMN `BaseElement` can be extended with additional attributes by associating the element with an `ExtensionDefinition`. At the same time, every extended BPMN element contains the actual extension attribute value. The attribute value, defined by the element `ExtensionAttributeValue` contains the value of type `Element` and has an association to the corresponding attribute definition.

In order to create a BPMN Extension for adding a fuzzy attribute, an XML schema, based on Complex Type, is defined for the type `tFuzzyAttribute`.

A fuzzy attribute has an element of type `tFuzzySet` defined by the set name (*LValue*), domain (*Universe*) and a membership function (type `tMembershipF`) with parameters, restricted to a list of potential forms (see listing 1).

```
<xsd:complexType name="tFuzzyAttribute">
   <xsd:sequence>
      <xsd:element name="LinguisticValue" type="tFuzzySet"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="tFuzzySet">
   <xsd:sequence>
      <xsd:element name="LValue" type="xsd:string"/>
      <xsd:element name="Universe" type="xsd:string"/>
      <xsd:element name="Function" type="tMembershipF"/>
      <xsd:element name="Parameter" type="sd:double" minOccurs="2"
      maxOccurs="4">
   </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="tMembershipF">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="triangular">
      <xsd:enumeration value="gauss">
      ....
      <xsd:enumeration value="psig">
   </xsd:restriction>
</xsd:simpleType>
```

**Listing 1.** XML schema for tFuzzyAttribute

As a concrete example, consider the extension of the BPMN element `DataObject` with an attribute of type `tProfile`, a complex structure defined, among others, by an element called *BurglaryTime* of type `tFuzzyAttribute`. The Extension XML schemas for `tProfile` and a simple XML instance are presented in listing 2 and 3. As a remark, the types `ExtensionAttributeDefinition` and `ExtensionAttributeValue` are not applicable when the XML schema interchange is used since the XSD mechanisms for supporting "AnyAttribute" and "Any" type already satisfy the requirements of defining new attributes/attribute values.

```
<xsd:complexType name="tProfile">
   <xsd:sequence>
      <xsd:element name="BurglaryTime" type="tFuzzyAttribute">
         ....
      </xsd:sequence>
      ...
</xsd:complexType>
```

**Listing 2.** Extension XML schema for DataObject

```
<bpmn:definitions id="ID_1" ...>
   ...
   <bpmn:extension mustUnderstand="true" definition="bpmn:tProfile"/>
   ...
   <bpmn:dataObject name="Profile" id="ID_5>
      <bpmn:BurglaryTime id="ID_6">
         <bpmn:LinguisticValue>
            <LValue "evening"/>
            <Universe "daily hours"/>
            <bpmn:Function>
               <value "triangular"/>
            </bpmn:Function>
```

```
          <Parameter> 17 20 22 24 </Parameter>
        </bpmn:LinguisticValue>
          ....
      </bpmn:BurglaryTime>
        ...
    </bpmn:dataObject>
    ...
</bpmn:definitions>
```

**Listing 3.** Sample XML instance

The BPMN Extension for fuzzy constraints is defined in a similar manner. The XML schema for `tFuzzyConstraint` type defines (see listing 4), among different elements, the fuzzy logical operators for aggregation, conjunction and disjunction, the defuzzification method and a sequence of `tFuzzyRule`. Each such rule defines an `ANDOperator`, `THENOperator`, `Weight` and the two main elements, `Premise` and `Consequence`. If `tConsequence` type contains one element of type `tFuzzyValue`, `tPremise` is defined as a list of `tFuzzyValue`-elements, each characterized by a fuzzy attribute (linguistic variable) and a fuzzy set (linguistic term).

```
<xsd:complexType name="tFuzzyConstraint">
    ...
    <xsd:element name="AggregationOperator" type="tAggregationOperator"/>
    <xsd:element name="ANDOperator" type="tANDOperator"/>
    <xsd:element name="THENOperator" type="tTHENOperator"/>
    <xsd:element name="Defuzzyfication" type="tDefuzzyfication"/>
      ...
      <xsd:complexType>
        <xsd:sequence minOccurs="1" maxOccurs="unbounded">
          <xsd:element name="Rule" type="tFuzzyRule"/>
        </xsd:sequence>
      </xsd:complexType>
   ...
</xsd:complexType>

<xs:complexType name="tFuzzyRule">
  <xsd:sequence>
    <xsd:element name="Weight" type="tWeight"/>
    <xsd:element name="Premise" type="tPremise"/>
    <xsd:element name="Consequence" type="tConsequence"/>
  </xsd:sequence>
</xsd:complexType>
  ...
<xsd:complexType name="tPremise">
  <xsd:sequence>
    <xsd:element name="Value" type="tFuzzyValue" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="tFuzzyValue">
  <xsd:sequence>
    <xsd:element name="LVariable" type="tFuzzyAttribute"/>
    <xsd:element name="LTerm" type="tFuzzySet"/>
  </xsd:sequence>
</xsd:complexType>
```

**Listing 4.** XML schema for FuzzyConstraint

An important remark is that a BPMN extension with a fuzzy constraint must be applied only on BPMN objects having an attribute of type `Expression`. Furthermore, BPMN 2.0 supports formal expressions, where the logic is captured in an executable form using a specified expression language.

## 7 Conclusions

Modelling inference structures that pertain to crime investigation rises necessarily the problem of handling the imperfect nature of forensic data and the uncertainties and conjectures characterizing crime analysis processes. Business Process Model and Notation, as the formalism used for modeling tasks (due to its intuitive usability and strong capability of representation) may accommodate the fuzzy inherent nature of crime processes using the vagueness of natural language descriptions/expressions. This article considers a new approach for the problem of fuzziness modeling: embedding fuzzy concepts into the BPMN formalism using the extensibility mechanism introduced in the current version of the BPMN specification.

This mechanism provides a set of extension elements, which allows BPMN adopters to attach additional attributes and elements to standard and existing BPMN elements, while keeping the validity (both syntactic and semantic) of standard diagrams. The fuzzy concepts introduced as extension (and defined as XML schemas) are *fuzzy attributes* (expressing linguistic variables taking as values fuzzy sets) and *fuzzy constraints* (fuzzy systems using a defuzzification method returning binary values).

Because the representation of fuzzy logic concepts as BPMN extensions was limited to particular cases (intended only as proof-of-concept), future research must concern the definition and validation of XML schemas supporting any possible fuzzy logic expression. The task of mapping fuzzy-based BPMN orchestration process models to WS-BPEL executable process models must be also approached. Furthermore this theoretical analysis must be done in parallel with the development of a dedicated modeling tool supporting fuzzy BPMN.

## References


1. Franke, K., Srihari, S.: Computational forensics: Towards hybrid-intelligent crime investigation. In: Information Assurance and Security. (2007) 383 –386
2. Cao, L.: Domain-driven data mining: Challenges and prospects. IEEE Transactions on Knowledge and Data Engineering **22** (2010) 755 – 769
3. Cotofrei, P., Stoffel, K.: Business process modelling for academic virtual organizations. In Boston, S., ed.: Pervasive Collaborative Networks. (2008) 213–220
4. Franke, K., Srihari, S.: Computational forensic: An overview. In: Computational Forensics. (2008)
5. Adam, O., Thomas, O.: A fuzzy based approach to the improvement of business processes. In: BPI'05 : Workshop on Business Process Intelligence. (2005) 25 – 35
6. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338 – 353
7. Zvieli, A., Chen, P.P.S.: Entity-relationship modeling and fuzzy databases. In: Int. Conf. on Data Engineering, IEEE Computer Society Press (1986) 320–327
8. Benedicenti, L., Succi, G., Vernazza, T., Valerio, A.: Object oriented process modeling with fuzzy logic. In: Proc. of ACM Symposium on Applied Computing. (1998) 267–271
9. Cox, E.: Striving for imprecision: Fuzzy knowledge bases for business process modeling. PC AI **13** (1999)

10. Cox, E.: Knowledge-based business process modeling: Complex systems design through a fusion of computational intelligence and object-oriented models. PC AI **16** (2002) 15 – 23
11. Völkner, P., Werners, B.: A simulation-based decision support system for business process planning. Fuzzy Sets and Systems **125** (2002) 275–288
12. Chen, S.M., Ke, J.S., Chang, J.F.: Knowledge representation using fuzzy petri nets. IEEE Transactions on Knowledge and Data Engineering **2** (1990) 311–319
13. Looney, C.: Fuzzy petri nets for rule-based decision making. IEEE Transactions on Systems, Man and Cybernetics **18** (1988) 178–183
14. Ye, Y., Jiang, Z., Diao, X., Du, G.: Extended event-condition-action rules and fuzzy petri nets based exception handling for workflow management. Expert Systems with Applications **38** (2011) 10847–10861
15. Becker, J., Rehfeldt, M., Turowski, K.: Auftragsabwicklung mit unscharfen daten in der industrie. In: Betriebliche Anwendungen von Fuzzy-Technologien. (1996) 51–61 (In German).
16. O. Thomas, T.D., Loss, P.: Towards enhanced business process models based on fuzzy attributes and rules. In: Proceedings of the Thirteenth Americas Conference on Information Systems, Keystone, Colorado, USA. (2007)
17. Thomas, O., Dollmann, T.: Fuzzy-EPC markup language: XML based interchange formats for fuzzy process models. Soft Computing in XML Data Management **255** (2010) 227–257
18. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. Information Sciences **8** (1975) 199–249
19. Klement, E., Mesiar, R., Pap, E.: Triangular norms. Kluwer (2000)
20. Yager, R.: On a general class of fuzzy connectives. Fuzzy Sets and Systems **4** (1980) 235 – 242
21. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
22. Sugeno, M.: Fuzzy measures and fuzzy integrals: a survey. Fuzzy Automata and Decision Processes (1977) 89 – 102
23. Fukami, S., Mizumoto, M., Tanaka, K.: Some considerations of fuzzy conditional inference. Fuzzy Sets and Systems **4** (1980) 243 – 273
24. Mamdani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies **7** (1975.) 1 – 13
25. Sugeno, M.: Industrial applications of fuzzy control. Elsevier Science (1985)
26. Boba, R.: Crime Analyses with Crime Mapping. Sage (2009)
27. Ribaux, O. et al. Intelligence-led Crime Scene Processing. Part II: Intelligence and Crime Scene Examination . Forensic Science International **199** (2010) 63–71
28. Ribaux, O. et al. Intelligence-led Crime Scene Processing. Part I: Forensic Intelligence. Forensic Science International **195** (2010) 10–16
29. Ribaux, O., Margot, P.: Case-Based Reasoning in Criminal Intelligence Using Forensic Case Data. Science & Justice **43** (2003) 135–143
30. Ribaux, O., Margot, P.: La trace comme vecteur d'information au service du renseignement. In: Traité de sécurité intérieure. Presses Polytechniques et Universitaires Romandes, Lausanne, Lausanne (2008) 300–321
31. Ribaux, O., Margot, P.: Inference Structures for Crime Analysis and Intelligence Using Forensic Science Data: the Example of Burglary. Forensic Science International **100** (1999) 193–210
32. OMG: http://www.omg.org/spec/BPMN/2.0/PDF. (2011)