



Modeling Uncertainty in Declarative Artifact-Centric Process Models

Rik Eshuis^(✉)  and Murat Firat 

School of Industrial Engineering, Eindhoven University of Technology,
Eindhoven, Netherlands
{h.eshuis,m.firat}@tue.nl

Abstract. Many knowledge-intensive processes are driven by business entities about which knowledge workers make decisions and to which they add information. Artifact-centric process models have been proposed to represent such knowledge-intensive processes. Declarative artifact-centric process models use business rules that define how knowledge experts can make progress in a process. However, in many business situations knowledge experts have to deal with uncertainty and vagueness. Currently, how to deal with such situations cannot be expressed in declarative artifact-centric process models. We propose the use of fuzzy logic to model uncertainty. We use Guard-Stage-Milestone schemas as declarative artifact-centric process notation and we extend them with fuzzy sentries. We explain how the resulting fuzzy GSM schemas can be evaluated by extending an existing GSM engine with a tool for fuzzy evaluation of rules. We evaluate fuzzy GSM schemas by applying them to an existing fragment of regulations for handling a mortgage contract.

1 Introduction

Many business processes are performed by knowledge workers, who are responsible for making decisions and adding information, for instance by analyzing data. These knowledge-intensive processes [6] are centered around certain key business entities to which information is added and about which decisions are made, for instance a production order or a request for quotes. Artifact-centric process models have been proposed to model such knowledge-intensive processes [18]. Business artifacts combine data and process aspects of key entities in a holistic way [14]. To allow flexibility, such knowledge-intensive processes are often modeled in a declarative way with rules [9], rather than a procedural way.

Business artifacts contain business rules that specify conditions under which actions are taken [5]. However, these rules are modeled in a classical way, assuming clear-cut boundaries between different states of the world. The variables in the rules may in reality have some degree of uncertainty or may not be described precisely. This is caused by either having insufficient data or lacking concrete definitions of the considered concepts. For example, is there a unique loan amount that distinguishes small loans and big loans? Or in case we only have a small sampling of the requested loan amounts, then how reliable will it be to use a crisp number as the boundary for small and big loans?

Several approaches for modeling uncertainty exist, notably probabilistic and fuzzy modeling [12]. Probabilistic models rely on frequency-based quantitative analysis while fuzzy models deal with uncertainties in a qualitative way, modeling the imprecise way of human reasoning [12, 15]. We propose to model the uncertainties in decision-intensive processes using fuzzy rather than probabilistic models since there is a lack of rich data to generate probabilistic models and a need to support the use of knowledge from human experts.

Fuzzy models describe the universe of discourse using *linguistic* terms with corresponding linguistic labels instead of specifying precise numerical values (including 0–1 binary representations) [12]. Each linguistic label has a membership function defined in the universe of discourse. These linguistic labels may be “large”, “high”, or “most”, depending on the context. Another important point of use of fuzzy modeling is to model the decision-making of knowledge workers which leads to the extraction of their domain expertise. Modeling expert knowledge properly is clearly beneficial to enable automated decisions of high quality.

For instance, suppose a loan is classified as large if it exceeds 100\$, and not large otherwise. In reality, it can be uncertain what exactly comprises the class of large loans. A loan of 99K\$ is not large according to the classification, but obviously “more” large than a loan of 10K\$. Knowledge workers may have their own understanding of when a loan is large or small, which fuzzy modeling can capture [15].

The goal of this paper is to explore the use of fuzzy modeling for declarative artifact-centric process models. We define how to “fuzzify” such artifact models by using fuzzy rules. As host notation for modeling declarative artifact-centric schemas, we use Guard-Stage-Milestone (GSM) schemas [11]. The language GSM schemas has inspired CMMN, the OMG standard on Case Management [4], so the results can provide a stepping stone for incorporating fuzzy reasoning in CMMN schemas. As we explain in Sect. 4, existing GSM engines can support fuzzy GSM schemas by invoking fuzzy rule evaluators.

The remainder of this paper is organized as follows. Section 2 introduces a running example that is revisited in the remainder of this paper. Section 3 defines GSM schemas and introduces fuzzy modeling. Section 4 explores how fuzzy modeling can be applied to GSM schemas, using the running example as illustration. Section 5 evaluates the use of fuzzy GSM schemas by modeling a fragment of a complex regulated process for handling mortgages. Section 6 discusses related work. Section 7 ends the paper with conclusions and an outlook for future work.

2 Running Example

We use a running example based on a fragment of a real-world process from an international technology company, in which business criteria for partner contracts are assessed [8]. The process fragment has the following behavior. First, data is gathered needed to perform the assessment. Next, three activities are

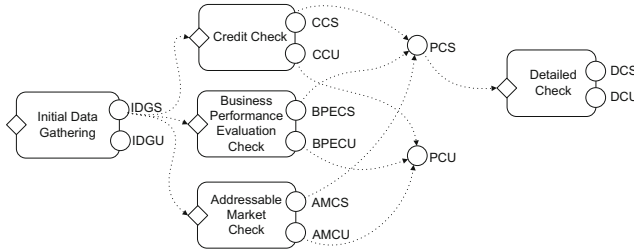


Fig. 1. Business Criteria Assessment process (BCA^{base}) [8]

Table 1. Stages and guards for BCA^{base} in Fig. 1

Stage	Guard
Initial Data Gathering	E:AssessmentRequest
Credit Check	IDGS
Business Performance Evaluation Check	$IDGS \wedge \text{employee_count} \geq 300$
Addressable Market Check	$IDGS \wedge \text{annual_revenue} \geq \$500K$
Detailed Check	PCS

performed in parallel as a pre-check. First, the credit is checked to ensure that the credit limit of the partner is still valid. Second, the past performance of the partner is evaluated and checked, but only if the partner has more than 300 employees. Third, the market addressed by the partner is assessed if the annual revenue exceeds \$500K. If the three parallel checks are successful, the pre-check succeeds and a detailed check is performed, which may either succeed or fail.

Figure 1 shows the lifecycle part of the GSM schema for this Business Criteria Assessment process BCA [8]. Rounded rectangles denote stages, in which work is performed. Circles denote milestones, which are business objectives achieved by completing the work in a stage to which a milestone is attached or by another external event. Diamonds denote guards, which specify conditions called sentries under which a stage opens. Sentries are also used for milestones to specify the conditions when they are achieved. Table 1 lists the sentries (guards) of the defined stages and Table 2 the sentries of those defined milestones that are revisited in the sequel of this paper; the full list for all milestones can be found elsewhere [8]. Dashed arrows in Fig. 1 denote dependencies between stages and milestones caused by sentries: for instance, the sentry (guard) of stage Detailed Check states that the stage is opened if milestone PCS has been achieved, so stage Detailed Check depends on milestone PCS.

The GSM schema in this paper is modeled in a crisp way: the sentries specify exact conditions for each stage to be opened and each milestone to be achieved. In reality, the knowledge worker may use more fluid conditions. In Sect. 4 we explore how fuzzy logic can model such conditions.

Table 2. Some milestones and their sentries for BCA^{base} in Fig. 1. ‘;’ separates different sentries

Milestone	Full name	Sentry
...
CCS	Credit Check Successful	$C:\text{Credit Check} \wedge \text{rating} \geq 8$
CCU	Credit Check Unsuccessful	$C:\text{Credit Check} \wedge \text{rating} < 8$
PCS	Pre-checks Successful	$CCS \wedge \text{BPECS} \wedge \text{AMCS} ;$ $CCS \wedge \text{employee_count} < 300 \wedge \text{AMCS} ;$ $CCS \wedge \text{BPECS} \wedge \text{annual_revenue} < \$500K ;$ $CCS \wedge \text{employee_count} < 300 \wedge$ $\text{annual_revenue} < \$500K$
PCU	Pre-checks Unsuccessful	$CCU \vee \text{BPECU} \vee \text{AMCU}$
..

3 Preliminaries

We introduce GSM schemas in more detail, as well as fuzzy modeling.

3.1 GSM Schemas

This section presents formal definitions for the variant of GSM schemas used in this paper. Given the focus on the integration of sentries with fuzzy modeling, we have chosen a lightweight GSM variant [8] in which there is no hierarchy and executions are monotonic, i.e., each stage and each milestone changes value only once. However, the classic GSM schemas [5] can be extended in a similar way. Given the restricted space, the presentation is concise, explaining those GSM details that are important to understand the remainder of this paper. For an extended introduction to classic GSM schemas, we refer the interested reader to other papers [5, 7, 11].

A GSM schema Γ consists of attributes, subdivided into data attributes, stage attributes and milestone attributes. The latter two represent the status of stages and milestones. If a stage attribute is true (false), the stage is open (closed). If a milestone attribute is true (false), the milestone is achieved (invalid). Each data attribute a has a type $\text{type}(a)$ which is scalar, e.g., string, character, integer, float, etc. Status attributes have type Boolean.

We assume a propositional condition language \mathcal{C} that includes fixed predicates over scalars (e.g., ‘ \leq ’ over integers or floats), and Boolean connectives. The condition formulas may involve stage, milestone, and data attributes. A status attribute will take the value *True* if one of its sentries goes true.

A *sentry* ψ defines a condition for a status attribute, i.e., a stage or milestone, to become true. A *sentry* ψ has one of the three forms: “ φ ”, “ $C:S$ ”, or “ $C:S \wedge \varphi$ ”, where φ is a condition formula ranging over the attributes of Γ . Here “ $C:S$ ” is called the *completion event* for stage S . Also, $C:S$ (if present) is the *completion event* for ψ and φ (if present) is the *formula* for ψ .

Definition 3.1. A GSM *schema* is a tuple $\Gamma = (\mathcal{A} = \mathcal{A}_D \cup \mathcal{A}_S \cup \mathcal{A}_m, \mathcal{E}, sen)$ where:

- \mathcal{A} is a finite set of attributes.
- \mathcal{A}_D is a finite set of data attributes.
- \mathcal{A}_S is a finite set of stage attributes.
- \mathcal{A}_m is a finite set of milestone attributes.
- $\mathcal{E} = \{ C:S \mid S \in \mathcal{A}_S \}$ is the set of stage completion events.
- The *sentry assignment* sen is a function from $\mathcal{A}_S \cup \mathcal{A}_m$ to sets of sentries with formulas in the condition language \mathcal{C} ranging over \mathcal{A} . Each element of set $sen(v)$ for $v \in \mathcal{A}_S \cup \mathcal{A}_m$ is called a *sentry* of v .

Sentries define dependencies between stage and milestone attributes of Γ . For $a_1, a_2 \in \mathcal{A}_S \cup \mathcal{A}_m$, a dependency (a_1, a_2) signifies that there is a sentry of a_2 that references a_1 . The dependencies are visualized as dashed arrows in Fig. 1.

Definition 3.2. For a GSM schema $\Gamma = (\mathcal{A}, \mathcal{E}, sen)$ a *snapshot* is a mapping σ from \mathcal{A} into values of appropriate type. For stage and milestone attributes, the only permitted values are *False* and *True*. Initially, all stage and milestone attributes are *False*.

Let $\Gamma = (\mathcal{A}, \mathcal{E}, sen)$ be a GSM schema. Let $\psi = C:S \wedge \varphi$ be a sentry for a status attribute $v \in \mathcal{A}_S \cup \mathcal{A}_m$, so $\psi \in sen(v)$. Given a snapshot σ of Γ and a stage completion event $C:S'$ that occurs, $\sigma \models \psi$ denotes that $C:S = C:S'$ and $\sigma \models \varphi$, so σ satisfies the formula φ of sentry ψ .

If in a snapshot a stage completion event occurs of an open stage, then several sentries may become satisfied. If the sentry of a status attribute becomes satisfied, the status attribute becomes true. Then sentries referencing the status attribute may also become true, which in turn means that the status attributes that own the sentries become true. For instance, stage *Credit Check* is open and milestones *BPECS* and *AMCS* have been achieved, then the completion event $C:\text{Credit Check}$ with payload $\langle \text{rating}, 9 \rangle$ results in achieving milestone *CCS*, which in turn leads to achieving milestone *PCS*, which in turn opens stage *Detailed Check*. In the resulting snapshot, no sentries are satisfied and the evaluation of sentries stops and the system waits for the next stage completion event of $C:\text{Detailed Check}$. Thus, the evaluation of sentries has a cascading effect, which is called a *Business-step*, or *B-step* for short [5].

To properly compute B-steps, dependency graphs are used [5]. For each completion event $C:S$ of stage S , a dependency graph $DG(C:S)$ is created whose nodes are $V = \mathcal{A}_S \cup \mathcal{A}_m$ and an edge (v_1, v_2) exists if there is a sentry of v_2 that references v_1 . If stage completion event $C:S$ occurs, then the sentries of v_1 must be evaluated before those of v_2 , in order to ensure that each change in status attribute is justified when examining the starting and ending snapshots of the B-step [5].

3.2 Fuzzy Modeling

In real-world decision making, obtaining precise parameters for decision models may be hardly possible or precise values for some threshold may not be meaningful. A classical example is assessing whether the weather is hot, cold or medium based on the temperature. The same value of temperature can result in different assessments for different countries. For instance a temperature of 30° is considered hot in the Netherlands, but medium in Turkey.

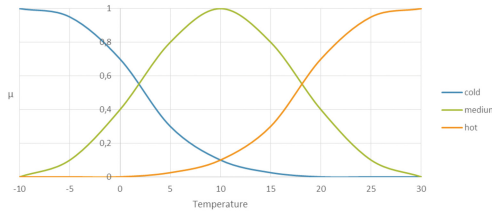


Fig. 2. Membership functions of linguistic terms of *Temperature*

Fuzzy logic has been proposed to capture this uncertainty in precision [15]. Fuzzy logic uses linguistic variables, i.e., variables whose values are different *linguistic terms* that are defined in terms of a base variable. For a linguistic variable x , each linguistic term t is defined in terms of a membership function $\mu_t : dom(X) \rightarrow [0, 1]$, which maps each value of X to the degree in which term t holds, zero meaning t is not applicable at, one that t is completely applicable. Figure 2 shows the membership functions for the linguistic terms “hot”, “medium”, “cold” for the linguistic variable *Temperature*. These definitions apply for countries like the Netherlands; for a country like Turkey, the membership functions would be different. For instance, $\mu_{hot,TR}(30) = 0.75 < \mu_{hot,NL}(30) = 1$, where TR represents Turkey and NL the Netherlands.

In fuzzy logic, logical statements contain fuzzy operands, indicating that uncertain or poor information is available. The two states of the satisfaction of a logical statement, namely *True* (1) and *False* (0), are extended to take value in the interval $[0, 1]$. This requires defining *fuzzy aggregation operators*, that are the fuzzy counterparts to logical operands like AND and OR.

Fuzzy Aggregation: The logical aggregation of AND and OR operators are generalized in fuzzy logic by using t-norm (conjunctive) and t-conorm (disjunctive) operators, respectively [12, 15]. These are most commonly used, although there are other aggregation operators, like compensative, non-compensative, and weighted operators. By definition, any conjunctive (disjunctive) operator should come up with one value that is not greater (smaller) than any individual value in the aggregation. So the highest (smallest) value is delivered in a conjunctive (disjunctive) aggregation, by *minimum* (*maximum*) operator. Besides these basic optimistic operators for an aggregation, other operators are proposed in the literature [12, 15]. In this paper, we use *product* operator for the conjunctive aggregation, and the influence of the aggregation operators on the decisions is not contained in the scope of our work.

4 Fuzzifying GSM Schemas

In this section we show how the GSM schema in Sect. 2 can be modeled using fuzzy logic, i.e., how the sentries of both stages and milestones can be fuzzified.

4.1 Linguistic Modeling of Variables

Artifact models are abstractions of real-world procedures that include uncertainty due to several reasons like insufficient sampling size, involvement of human perception and so on. Thus, in classic GSM schemas values of crisp variables can actually carry considerable uncertainty. These variables can therefore be better modeled as linguistic variables with various linguistic labels such as “low”, “medium”, “high”, and so on. The range of values for each linguistic label may vary across different organizational units, hence it is usually not easy, perhaps even impossible, to accurately capture these different states in a quantitative manner.

First, let us consider the relation between number of employees and company size. In the crisp GSM schema in Sect. 2, the sentry “`employee_count ≥ 300`” is used to *classify a company as big*, if the result of this check is true. Now, 300 is a somewhat arbitrary number and specified by a human expert as the threshold value. However, a razor sharp classification of a company to be small or big may not be a wise thing to do. For the sake of extracting more knowledge from an expert, the following question can be asked: “*For what employee numbers is a company considered to be big to what extent?*”.

Then the answer will lead us to define the membership function in part (b) of Fig. 3. Consequently, the binary check “`employee_count ≥ 300`” is replaced with the fuzzy statement “*the company is big*” with a given degree of correctness $\mu_{big}(\text{employee_count}) \in [0, 1]$, a so-called membership value that can be read from part (b) of Fig. 3.

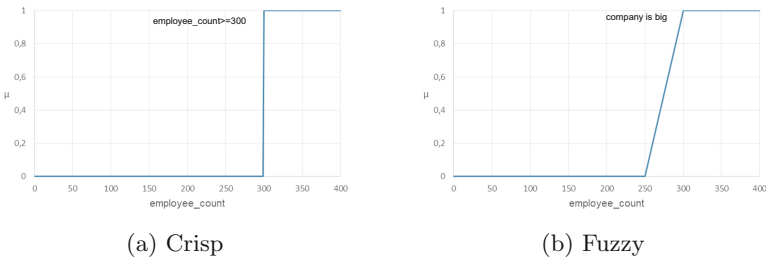


Fig. 3. Binary and fuzzy logic for employee count check

We define a procedure to fuzzify the crisp variables of a given GSM schema:

- 1: Pick a crisp variable.
- 2: Find different numerical checks involving that crisp variable.
- 3: Interpret every check linguistically.
- 4: Define membership functions of the linguistic labels based on either expert knowledge or a limited sample, if no expert knowledge is available.

Let us apply the above procedure for another crisp variable in our GSM example of Fig. 1.

- Crisp variable: `annual_revenue`
- Numerical checks: `annual_revenue` \geq \$500K
- Linguistic interpretation: “company’s financial size is *big*”
- Membership function: $\mu_{big}(ar) = \begin{cases} 0, & \text{if } ar \leq 400 \\ \frac{ar-400}{100}, & \text{if } 400 \leq ar < 500 \\ 1, & \text{if } 500 \leq ar, \end{cases}$

where $ar = \text{annual_revenue}$.

4.2 Condition Formula of Sentries with Fuzzy Aggregation

We explain how to compute the truth value of a sentry with linguistic variables.

First, we show how to write the completion event of a stage in the fuzzy notation. Practically, we keep the completion event crisp, but the notation is adapted to have a single consistent notation for fuzzy sentries. The completion event of stage S has the corresponding fuzzy linguistic statement “Completion event of Stage S has *occurred*”. Then we define the discrete fuzzy membership function for the linguistic label “*occurred*” as

$$\mu_{occurred}(C:S) = \{(True, 1), (False, 0)\}.$$

The reason why we interpret the completion event in a binary fashion is that there is no uncertainty regarding the completion of a stage: a stage has either completed or not. However, it can make sense to allow stages to gradually complete. We plan to study gradual completion of stages in future research.

In the classical GSM schemas, sentries may contain logical “and” and “or” operators in their condition formula. In our fuzzy modeling of GSM schema, condition formulas are computed using fuzzy aggregations, introduced in Sect. 3.2. For example, the sentry “C:Credit Check \wedge rating \geq 8” of milestone CCS in Table 2 can be transformed into

“C:Credit Check has *occurred*” and “rating is *high*”

where `rating` is modeled as linguistic variable with a linguistic label “high” that can be interpreted as “being approximately greater than or equal to 8”.

Computation of a fuzzy aggregation is performed by using conjunctive and disjunctive operators, as mentioned in Sect. 3.2. So the condition formula is computed as

$$\mu_{CCS} = \mu_{occurred}(C : \text{CreditCheck}) \times \mu_{high}(r)$$

where r denotes the crisp value of rating, and the “product” t-norm operator is used in the above conjunctive fuzzy aggregation.

Note that condition formulas may also contain the satisfaction value of other condition formulas as a component in the aggregation. For example, the fuzzification of the sentry “ $CCS \wedge BPECS$ ” of the milestone PCS is done as

$$\mu_{PCS} = \mu_{CCS} \times \mu_{BPECS}$$

where again the “product” t-norm operator is used.

Given a stage or milestone $a \in \mathcal{A}_S \cup \mathcal{A}_m$, evaluating a fuzzy sentry $\varphi \in sen(a)$ results in a fuzzy value between 0 and 1. Defuzzification is needed to derive a crisp Boolean value that denotes whether a becomes true or value. Assuming a general threshold value $\alpha \in [0..1]$ for the entire GSM schema, we define a defuzzification function that assigns a status attribute true if its fuzzy value $\mu(a)$ exceeds the threshold α :

$$defuzzify(a) = \begin{cases} 1, & \text{if } \mu(a) \geq \alpha \\ 0, & \text{otherwise.} \end{cases}$$

4.3 Incorporating Fuzzy Sentries in the GSM Execution

So far, we have presented how sentries can be fuzzified. We now sketch a potential solution how the evaluation of fuzzy sentries can be done at run time (cf. Fig. 4). The only difference with the regular GSM semantics, in which sentries are evaluated to perform a B-step (see Sect. 3.1) is that sentries are now evaluated using fuzzy reasoning, consisting of three steps that are standard in evaluating fuzzy rules [15]: fuzzify the crisp input to determine the values of the membership functions of the fuzzy variables referenced in the sentry, then evaluate the sentry, and next defuzzify the input by applying the change for which the sentry is a condition, if the sentry exceeds the threshold value. The functions used in these steps have been defined above.

For classical GSM schemas, the only change is that sentries are evaluated using classical logic [5], so (de)fuzzification does not apply. Thus, the proposed extension is conservative: GSM engines only need to invoke a fuzzy evaluation tool that implements the (de)fuzzification and aggregation functions defined above.

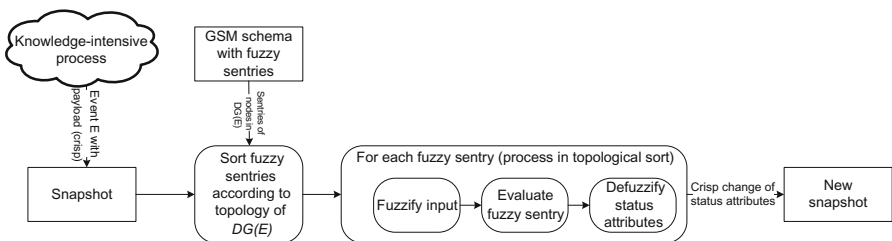


Fig. 4. Evaluation of fuzzy sentries in GSM execution

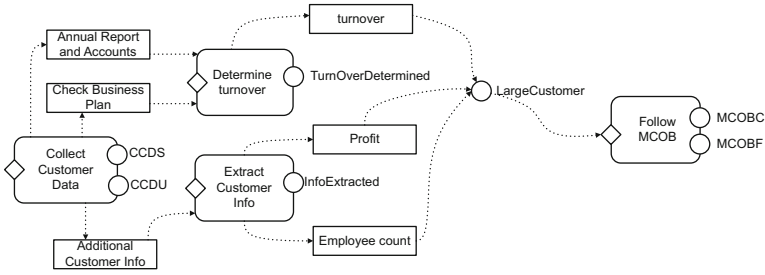


Fig. 5. Part of GSM schema for processing regulated mortgage contracts

5 Evaluation

As a first evaluation step, we apply the approach to some of the rules defined by the Financial Conduct Authority in the UK as part of the Mortgage Code of Business (MCOB) for financial firms that offer regulated mortgage contracts: www.handbook.fca.org.uk/handbook/MCOB. Below we show a few MCOB regulations for regulated mortgage contracts that financial firms use to decide whether they have to adhere to the code of business for the customer. We show how they are formalized with the fuzzy modeling approach advocated in this paper.

1.2.3. *In relation to a regulated mortgage contract for a business purpose (1) MCOB applies if the customer is not a large business customer; and (2) if MCOB applies, a firm must comply with MCOB in full, taking into account tailored exceptions and provisions.*

1.2.6. *In determining whether a customer is a large business customer, a firm will need to have regard to the figure given for the customer’s annual turnover in the customer’s annual report and accounts or business plan. In addition, a firm may rely on information provided by the customer about the annual turnover, unless, taking a common-sense view of this information, it has reason to doubt it.*

Figure 5 shows part of a fuzzy GSM schema that operationalizes these regulations. As additional customer info, we consider the profit and the number of employees an organization has. First, we define the linguistic terms and corresponding labels in Table 3. Then for every linguistic label a membership function is defined based on expert/domain knowledge.

The most important element, milestone **LargeCustomer**, has two fuzzy sentries:

- Business sales volume is high
- Business sales volume is high *and* Customer financial situation is good *and* Customer team size is big

Table 3. Converting crisp variables to linguistic terms and labels

Crisp variable	Linguistic term	Label	Membership function
turnover	Business sales volume	high	$\mu_{high}(\text{turnover})$
profit	Customer financial situation	good	$\mu_{good}(\text{profit})$
employee count	Customer team size	big	$\mu_{big}(\text{employee count})$

The second sentry requires a three-component fuzzy aggregation:

$$\mu_{\text{LargeCustomer}} = \mu_{high}(\text{turnover}) \times \mu_{good}(\text{profit}) \times \mu_{big}(\text{employee count})$$

where the product operator is used for the t-norm operation. Next, defuzzification as defined below results in an binary output of the milestone `LargeCustomer` which directly determines if the stage `Follow MCOB`, which has sentry “`¬LargeCustomer`” turns to *True*.

$$\text{defuzzify}(\text{LargeCustomer}) = \begin{cases} 1, & \text{if } \mu_{\text{LargeCustomer}} \geq \alpha \\ 0, & \text{otherwise,} \end{cases}$$

where α is the threshold introduced in Sect. 4.

In a classical GSM schema, the decision of when a customer is large is naturally modeled by having an atomic stage, that has the relevant data attributes as input, and upon completion achieves either milestone `LargeCustomer` or `NonLargeCustomer`. The actual decision logic is then hidden in the atomic stage. Using fuzzy logic, the reasoning behind the decision can be represented in sentries, so the fuzzy GSM model forces to make the reasoning rules explicit.

This first evaluation step shows that fuzzy GSM schemas can be used to represent uncertainty that is inherent in real-world regulations. We plan to evaluate the approach in industrial case studies.

6 Related Work

Fuzzy modeling has been applied to many different application domains, such as control and decision making and optimization [15]. However, only a few papers have applied fuzzy modeling in the context of BPM.

Thomas et al. [1, 17] explore the combination of fuzzy modeling and Event-driven Process Chains (EPCs). They advocate that the resulting fuzzy workflows replace the classical, crisp workflows. Ye et al. [20] propose the use of fuzzy logic to handle exceptions in workflow management, using fuzzy Petri nets as underlying model. They focus on extending existing crisp workflow models and crisp workflow systems to incorporate fuzzy reasoning. In contrast to these works [1, 17, 20], we propose that the fuzzy sentries are only used as front end to decide the truth value of sentries, so a non-fuzzy GSM engine is still used to drive the execution. Furthermore, these works consider procedural process models, whereas we consider declarative, artifact-centric process models.

Fuzzy logic has also been applied to represent uncertainty for simulation [2, 19] and analysis [21] of business processes. Landry et al. [13] apply fuzzy logic to distributed workflows to support collaborative crisis management under uncertainty. Slavíček [16] presents a fuzzy ontology-based workflow system. The focus is on integrating fuzzy workflow systems with fuzzy ontologies. None of these papers focuses on declarative, artifact-centric processes.

In the broader area of BPM, fuzzy modeling has been applied as well. For instance, Hakim et al. [10] present a methodology based on fuzzy logic to select processes for reengineering while Bazhenova et al. [3] define how to extract fuzzy decision models from event logs. However, in these papers the processes themselves are not modelled in a fuzzy way.

7 Conclusion

We have proposed fuzzy GSM schemas as a technique for modeling uncertainty in declarative artifact-centric process models. The added value of fuzzy GSM schemas over classical GSM schemas is twofold. Firstly, they are capable of capturing expert knowledge via defining rules in linguistic terms that match human understanding of the problem domain better than the rules in classical GSM schemas. Secondly, fuzzy GSM schemas are more useful for the cases where it is hard, even impossible, to collect enough information to make concrete decisions while performing the business process. The resulting uncertainty is better captured using linguistic variables in decision making, as done in fuzzy models, rather than relying on crisp variables as used in classical GSM schemas. We also discussed how existing GSM engines can support fuzzy GSM schemas by invoking fuzzy rule evaluators. Thus, the proposed GSM extension is lightweight and does not require any redefinition of existing GSM engines.

For future work, we plan to explore the use of fuzzy stages and milestones for fuzzy GSM schemas. This also will affect the execution of GSM schemas, i.e., the notion of Business step needs to be adjusted accordingly. Moreover, we plan to incorporate fuzzy inference systems into fuzzy GSM schemas involving highly complicated decision making parts. Finally, we plan to apply the approach in several case studies in organizations.

References

1. Adam, O., Thomas, O., Martin, G.: Fuzzy workflows - enhancing workflow management with vagueness. In: Proceedings of the EURO/INFORMS 2003 (2003)
2. Azadeh, A., Haghnevis, M., Khodadadegan, Y., Madadi, M.: Modeling and improvement of the integrated business and production processes by fuzzy simulation. In: Proceedings of the SpringSim 2009. SCS/ACM (2009)
3. Bazhenova, E., Haarmann, S., Ihde, S., Solti, A., Weske, M.: Discovery of fuzzy DMN decision models from event logs. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 629–647. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_39

4. BizAgi, et al.: Case Management Model and Notation (CMMN), v1.1, OMG Document Number formal/2016-12-01, Object Management Group (2016)
5. Damaggio, E., Hull, R., Vaculín, R.: On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles. *Inf. Syst.* **38**, 561–584 (2013)
6. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: characteristics, requirements and analysis of contemporary approaches. *J. Data Semant.* **4**(1), 29–57 (2015)
7. Eshuis, R., Hull, R., Sun, Y., Vaculín, R.: Splitting GSM schemas: a framework for outsourcing of declarative artifact systems. *Inf. Syst.* **46**, 157–187 (2014)
8. Eshuis, R., Hull, R., Yi, M.: Property preservation in adaptive case management. In: Barros, A., Grigori, D., Narendra, N.C., Dam, H.K. (eds.) *ICSOC 2015*. LNCS, vol. 9435, pp. 285–302. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48616-0_18
9. Goedertier, S., Vanthienen, J., Caron, F.: Declarative business process modelling: principles and modelling languages. *Enterp. IS* **9**(2), 161–185 (2015)
10. Hakim, A., Gheitasi, M., Soltani, F.: Fuzzy model on selecting processes in business process reengineering. *Bus. Proc. Manag. J.* **22**(6), 1118–1138 (2016)
11. Hull, R., et al.: Introducing the guard-stage-milestone approach for specifying business entity lifecycles. *Proc. WS-FM* **2010**, 1–24 (2010)
12. Klir, G.J.: *Uncertainty and Information: Foundations of Generalized Information Theory*. Wiley, Hoboken (2006)
13. Landry, J.-F., Ulmer, C., Gomez, L.: Fuzzy distributed workflows for crisis management decision makers. In: Ortiz-Arroyo, D., Larsen, H.L., Zeng, D.D., Hicks, D., Wagner, G. (eds.) *EuroISI 2008*. LNCS, vol. 5376, pp. 226–236. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89900-6_23
14. Nigam, A., Caswell, N.S.: Business artifacts: an approach to operational specification. *IBM Syst. J.* **42**(3), 428–445 (2003)
15. Ross, T.J.: *Fuzzy Logic with Engineering Applications*. Wiley, Chichester (2010)
16. Slavíček, V.: An ontology-driven fuzzy workflow system. In: van Emde Boas, P., Groen, F.C.A., Italiano, G.F., Nawrocki, J., Sack, H. (eds.) *SOFSEM 2013*. LNCS, vol. 7741, pp. 515–527. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35843-2_44
17. Thomas, O., Dollmann, T., Loos, P.: Rules integration in business process models - a fuzzy oriented approach. *Enterp. Model. Inf. Syst. Arch.* **3**(2), 18–30 (2008)
18. Vaculín, R., Hull, R., Heath, T., Cochran, C., Nigam, A., Sukaviriya, P.: Declarative business artifact centric modeling of decision and knowledge intensive business processes. *Proc. EDOC* **2011**, 151–160 (2011)
19. Völkner, P., Werners, B.: A simulation-based decision support system for business process planning. *Fuzzy Sets Syst.* **125**(3), 275–287 (2002)
20. Ye, Y., Jiang, Z., Diao, X., Du, G.: Extended event-condition-action rules and fuzzy Petri nets based exception handling for workflow management. *Expert Syst. Appl.* **38**(9), 10847–10861 (2011)
21. Zakarian, A.: Analysis of process models: a fuzzy logic approach. *Int. J. Adv. Manuf. Technol.* **17**(6), 444–452 (2001)