


MODELFY: A Model-driven Solution for Decision Making based on Fuzzy Information


María Castañeda

(Universidad Complutense de Madrid, Madrid, Spain
mcasta01@ucm.es)

Mercedes G. Merayo

(Universidad Complutense de Madrid, Madrid, Spain
 <https://orcid.org/0000-0002-4634-4082>, mgmerayo@fdi.ucm.es)

Juan Boubeta-Puig

(University of Cadiz, Cádiz, Spain
 <https://orcid.org/0000-0002-8989-7509>, juan.boubeta@uca.es)

Iván Calvo

(Universidad Complutense de Madrid, Madrid, Spain
ivcalvo@ucm.es)

Abstract: There exist areas, such as the disease prevention or inclement weather protocols, in which the analysis of the information based on strict protocols require a high level of rigor and security. In this situation, it would be desirable to apply formal methodologies that provide these features. In this scope, recently, it has been proposed a formalism, *fuzzy automaton*, that captures two relevant aspects for fuzzy information analysis: imprecision and uncertainty. However, the models should be designed by domain experts, who have the required knowledge for the design of the processes, but do not have the necessary technical knowledge. To address this limitation, this paper proposes MODELFY, a novel model-driven solution for designing a decision-making process based on fuzzy automata that allows users to abstract from technical complexities. With this goal in mind, we have developed a framework for fuzzy automaton model design based on a Domain-Specific Modeling Language (DSML) and a graphical editor. To improve the interoperability and functionality of this framework, it also includes a model-to-text transformation that translates the models designed by using the graphical editor into a format that can be used by a tool for data analysis. The practical value of this proposal is also evaluated through a non-trivial medical protocol for detecting potential heart problems. The results confirm that MODELFY is useful for defining such a protocol in a user-friendly and rigorous manner, bringing fuzzy automata closer to domain experts.

Keywords: Model-driven engineering, Fuzzy automata, DSML, Graphical modeling editor, Decision making, Support Diagnostic

Categories: D.2.2, D.2.3, F.4.3

DOI: 10.3897/jucs.76030

1 Introduction

Formal methods allow us to model processes with mathematical precision and rigor. Therefore, the development of methodologies based on formal approaches provides a

high degree of reliability for final products. There exist areas, for example medicine or weather events, in which the systems that are used, the protocols that are applied and the analysis of the information that is managed, require a high level of rigor and security. Hence, it would be desirable the use of formal methodologies in both the design of the corresponding models and the establishing of the rules for information analysis. One of the fields in the health area in which formal system development is beginning to be relevant is the disease prevention. The systems used for early diagnosis require to accurately process and analyze patient data, even in stages where they are still asymptomatic. In this scope, recently, it has been proposed a framework for information analysis [Calvo et al., 2018, Calvo et al., 2019] that considers two aspects very relevant in the analysis of clinical data: imprecision and uncertainty. In this work, the authors present a fuzzy version of automata that captures these characteristics. On the one hand, fuzzy automata (FZA) allow us to establish certain tolerance levels during the analysis of the data. On the other hand, it is possible to define restrictions that should be taken into account in determining the degree of confidence in the decisions made during the diagnostic process. This formalism also facilitates the representation of the correlation that must exist between the values of the parameters of interest for the detection of the pathology under consideration. This incorporates flexibility in the process, avoiding static analyses that can lead to wrong diagnoses. All these aspects make this formalism very suitable for use in the development of support systems for the early diagnosis of diseases, based on expert knowledge and the analysis of the data obtained from medical tests. Nevertheless, there are other domains in which the features provided by this formalism fits. For example, the design of protocols based on data analysis for severe weather events, identification of fire hazards etc. It is worth noting that the models should be designed by domain experts, who have the required knowledge for the design of the processes, but do not have the necessary technical knowledge. The tools used in the formal modeling frameworks require technical skills that experts have not acquired. This limitation may reduce the use of these tools. Therefore, we consider it would be useful to provide experts with a graphical tool that allows them to abstract from technical complexities and design models in a simple way. With this goal in mind, we have developed a framework for FZA model design based on a Domain-Specific Modeling Language (DSML) and a graphical editor: MODELIFY.

To improve the interoperability and functionality of this framework, it will be used with a transformation process that translates the models designed by using the graphical editor into a format that can be used by a tool for data analysis. Therefore, we decided to develop a model-to-text transformation that automatically translates graphical FZA models into code that can be executed by the AUNTY tool [Calvo et al., 2018]. AUNTY automatically analyses data using FZA models. The tool is very powerful but the automata to be used to process the data have to be modeled in a complex and not intuitive way. The idea is to provide our graphical editor with a functionality that performs the transformation of the models into the code that AUNTY requires for their execution.

The contributions of this paper are:

- Design of a DSML for modeling FZA.
- Development of a graphical modeling editor that allows the graphical design and automatic validation of models that conform the FZA metamodel proposed in our DSML.
- Design and implementation of a model-to-text transformation process for automatically generating FZA code executable on the AUNTY tool for data analysis.

The rest of the paper is structured as follows. Section 2 describes the main concepts of Model-Driven Engineering (MDE) and the FZA definition. Section 3 describes our model-driven approach in a nutshell, being detailed in depth in the following sections. Section 3 introduces the FZA DSML and Section 5 the graphical editor. Section 6 presents our case study, which consists in modeling the behavior of the heart using our framework and translating the model for being used in AUNTY. Section 8 details the most relevant related work. Finally, Section 9 highlights our conclusions and some lines for future work.

2 Preliminares

In this section, the main concepts of MDE and FZA are explained.

2.1 Model-driven Engineering

MDE [Brambilla et al., 2017] is a discipline of software engineering, which deals with the use of models to improve productivity, quality and software maintenance, among others aspects, especially when it presents a high complexity, by increasing the level of abstraction and automation. MDE can be applied to different areas, from the development of new applications to reengineering based on models [Groenewegen and Visser, 2013, Popovic et al., 2015, Perić et al., 2014, Boubeta-Puig et al., 2015, Valero et al., 2021]. In this context, a system is defined as “*a generic concept for the design of an application, platform, or software device*” [Rodrigues da Silva, 2015]. A system can be composed of or related to other subsystems. A model is a simplified representation of a system, an abstraction, which allows us to understand it better. This representation can be textual or graphical and is created by a DSML. Three components are distinguished in the design process. Firstly, an abstract syntax composed of a metamodel and a set of validation rules that allow to determine if a model is well defined. Secondly, a specific syntax including information on how to represent (visually or textually) the concepts of the abstract syntax. Finally, a transformation process between models that allows to automate the software development, the redesign of software, its simulation, etc. Next we review the main concepts and definitions underlying the scope of MDE.

There are numerous model definitions [Seidewitz, 2003, Selic, 2003, Kühne, 2006] although there is consensus that a model is a simplified representation of a system. There are some properties that characterize the models [Ludewig, 2003]. A model makes it possible to identify the elements of the reality that it represents. In addition, a model must be useful, that is, it must serve a final purpose. Finally, it must be a simplified version of that reality, in which they do not have to be represented all aspects of it, only those relevant to the purpose of the model.

A model is represented according to a modeling language, whose abstract syntax is defined by a metamodel. Intuitively, a metamodel is a representation of all the models that can be expressed with that language, a model of models. The modeling language of metamodels is a meta-model. Based on this, it can define models that describe metamodels and other models that describe meta-models. Thus, there could be infinite levels of metamodeling abstraction, but it has been seen that meta-models can be defined based on themselves. The possibility that these languages can be defined with respect to themselves allows to have a finite architecture. Therefore, there is no need for languages to define meta-metamodels. Figure 1 presents the four-level architecture (M_0, M_1, M_2, M_3) from the Object Management Group (OMG) standards which is used to describe the

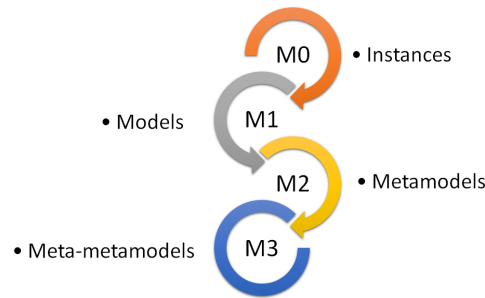


Figure 1: Four-layer metamodelling architecture

hierarchy of metamodeling. In this hierarchy it is assumed that an element of the level M_i conforms to the model of the level M_{i+1} which is instanced. In the case of the $M3$ level, meta-metamodels are self-conforming. The $M3$ level corresponds to the meta-metamodel. This layer provides a language for specifying metamodels, i.e. modeling languages. The meta-metamodels are defined on the basis of themselves, i.e. they conform to themselves. The standard metamodeling language proposed by the OMG is the Meta Object Facility (MOF) [Object Management Group, 2021]. In addition to this standard, other metamodel frameworks have been proposed such as the Eclipse Modeling Framework (EMF) [Steinberg et al., 2008] which has its own language of metamodeling, Ecore. In the $M2$ level are the metamodels defined according to a meta-metamodel. Metamodels basically constitute the definition of a modeling language, i.e. the concepts of the language and the relationships between them, as well as the rules to determine whether a model is well formed. A metamodel can look like a description of the set of all models that can be represented by that language [Brambilla et al., 2017]. Both Unified Modeling Language (UML) [Object Management Group, 2017] and Common Warehouse Metamodel (CWM) [Object Management Group, 2003] are examples of meta-metamodels that conform to the MOF meta-model. Level $M1$ corresponds to the models defined according to a metamodel. The models define concepts of the domain of interest based on concepts of the language defined by a metamodel. In level $M0$ we find the instances of the models that correspond to real entities of the domain, such as concepts, processes, etc.

Models must conform to the corresponding metamodel, which, as already mentioned, represents the abstract syntax of the modeling language. This syntax describes the structure of language, regardless of a specific notation. The concrete syntax of the language provides a representation of the modeling language that will allow to design models conform to the metamodel. This syntax can be graphical, such as that associated with UML, or textual, such as *Emfatic* [Daly, 2004] which defines *Ecore*. A graphical syntax presents graphical symbols, figures, labels and composition rules of such graphic elements. The graphical syntax must also provide the correspondence of the graphic symbols and the elements of the abstract syntax. A textual syntax includes elements to represent the information collected in the models, as well as keywords for represent the elements, limiting characters, etc... The modeling languages can be both general purpose languages and DSMLs. The latter are modeling languages designed specifically for a certain domain that includes its main primitives and abstractions [Voelter, 2013]. As examples, BPMN [Dumas et al., 2018, Object Management Group, 2020] for business process modeling and WebML [Ceri et al., 2000] for web application modeling or PGA

for modeling business strategies with the internal infrastructure and processes [Roelens and Bork, 2020].

2.2 Fuzzy Automata

In this work, finite automata will be used for process modeling of systems that require to consider certain degree of imprecision and uncertainty in the analysis of information. To deal with these features we have decided to rely on a formalism recently proposed by [Calvo et al., 2019], FZA, which allows to take into account these characteristics through the use of fuzzy logic. Intuitively, fuzzy logic allows to define relationships that provide a measure, in a range from 0 to 1, of the level of confidence in the analysis of data.

In order to determine this level of confidence *fuzzy relations* are used. Unlike the classic relations, whose evaluation return either 0 (true) or 1 (false), fuzzy relations provide a value in the interval $[0, 1]$. These relationships are parameterized by a value, denoted by δ , which indicates the maximum deviation allowed from the expected value. For example, in the case of evaluating whether $a \in [\alpha, \beta]$ and such value is in the interval, the relationship is met with confidence 1. However, if $a < \alpha$ or $a > \beta$ and the distance is less than δ we will obtain a positive confidence, which decreases as this distance increases. Finally, if $a < \alpha + \delta$ or $a > \beta + \delta$ then the confidence level will be 0.

Definition 1 A fuzzy relation is a function $fr : \mathbb{R}^n \rightarrow [0, 1]$ where $n \geq 2$.

Initially, in this framework, we have only considered linear relations because we think they are more intuitive than non-linear relations. Nevertheless, it could be easily extended for dealing with any other kind of fuzzy relations. In the following we present the ones that we consider in this work.

Definition 2 Let $\delta \in \mathbb{R}_+$. We define the three relations $\overline{x \leq y}^\delta, \overline{x \geq y}^\delta, \overline{x = y}^\delta : \mathbb{R}_+^2 \rightarrow [0, 1]$ as follows:

$$\overline{x \leq y}^\delta \equiv \begin{cases} 1 & \text{if } x < y \\ \frac{\delta + y - x}{\delta} & \text{if } y \leq x \leq y + \delta \\ 0 & \text{if } y + \delta < x \end{cases}$$

$$\overline{x \geq y}^\delta \equiv \begin{cases} 1 & \text{if } x > y \\ \frac{\delta + x - y}{\delta} & \text{if } y \geq x \geq y - \delta \\ 0 & \text{if } y - \delta > x \end{cases}$$

$$\overline{x = y}^\delta \equiv \begin{cases} 0 & \text{if } x \leq y - \delta \\ \frac{x - y + \delta}{\delta} & \text{if } y - \delta < x \leq y \\ \frac{-x + y + \delta}{\delta} & \text{if } y < x \leq y + \delta \\ 0 & \text{if } y + \delta < x \end{cases}$$

In addition to the previous fuzzy relations, we introduce a ternary fuzzy relation $\overline{x \leq y \leq z}^\delta : \mathbb{R}_+^3 \rightarrow [0, 1]$ as follows

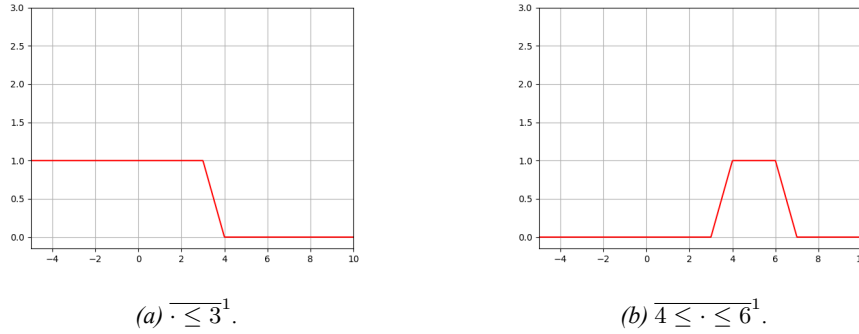


Figure 2: Fuzzy relations examples

$$\overline{x \leq y \leq z}^\delta \equiv \begin{cases} 0 & \text{if } z < x \\ 0 & \text{if } y \leq x - \delta \\ \frac{y-x+\delta}{\delta} & \text{if } x - \delta \leq y \leq x \leq z \\ 1 & \text{if } x < y < z \\ \frac{-y+z+\delta}{\delta} & \text{if } x \leq z \leq y \leq z + \delta \\ 0 & \text{if } z + \delta < y \end{cases}$$

Figures 2a and 2b present examples of the use of these relations in which we consider $\delta = 1$. The x-axis corresponds to the parameter while the y-axis indicates the confidence. A *fuzzy constraint* is a formula combining fuzzy relations.

Definition 3 Given a set X of variables, the set \mathcal{C}_X of fuzzy constraints over X is defined by the following EBNF:

$$\begin{aligned} C &::= True \mid F \\ F &::= F, F \mid fr \end{aligned}$$

where fr is a fuzzy relation.

The combination of the confidence values obtained from the evaluation of the fuzzy relations is performed by means of *t-norms*. A t-norm is a binary operation that allows to generalize the conjunction of propositional logic. A fuzzy constraint may include free variables. The evaluation of the fuzzy constraints corresponds to a *degree of constraint satisfaction* in the range from 0 to 1.

Definition 4 A t-norm is a function $\Delta : [0, 1] \times [0, 1] \rightarrow [0, 1]$ satisfying the following properties:

- *Commutativity*: $\forall x, y \in [0, 1]: x \Delta y = y \Delta x$.
- *Associativity*: $\forall x, y, z \in [0, 1]: (x \Delta y) \Delta z = x \Delta (y \Delta z)$.
- *Monotonicity*: $\forall x_1, x_2, y_1, y_2 \in [0, 1]: \text{if } x_1 \leq y_1 \wedge x_2 \leq y_2 \text{ then } x_1 \Delta x_2 \leq y_1 \Delta y_2$.
- *Identity element*: $\forall a \in [0, 1]: 1 \Delta a = a$.

Since t -norms are commutative and associative, we will sometimes use $\Delta(t_1, t_2, \dots, t_s)$ as a shorthand of $\Delta(t_1, \Delta(t_2, \dots, t_s) \dots)$. Abusing the notation, given two fuzzy constraints, $C_1 = fr_{11}, fr_{12}, \dots, fr_{1k}$ and $C_2 = fr_{21}, fr_{22}, \dots, fr_{2s}$, we will use $\Delta(C_1, C_2)$ to denote $\Delta(fr_{11}, fr_{12}, \dots, fr_{1k}, fr_{21}, fr_{22}, \dots, fr_{2s})$.

We can find different t -norms in the literature, whose application will depend on the nature of the data analysed, such as Gödel, Hamacher, Lukasiewicz or Frank [Kauers et al., 2011, Borgwardt et al., 2017]. In this framework, we will use the Gödel and the Hamacher t -norms which are defined as $G(x, y) = \min\{x, y\}$ and $H(x, y) = \frac{x \cdot y}{x + y - x \cdot y}$ for all $x, y \in [0, 1]$, respectively.

Definition 5 Given a set X of variables, a transformation function for a variable $x \in X$ is a function $f_x : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ that modifies the value of the variable x . The set of transformation functions, denoted \mathcal{TV} , is the set of all the total functions from \mathbb{R}_+ to \mathbb{R}_+ .

A valuation over X is a function that assigns a non-negative real number to every variable in X . The set of valuations of X , denoted \mathcal{V}_X , is the set of all the total functions from X to \mathbb{R}_+ . Given a valuation $vl \in \mathcal{V}_X$, a tuple of variables $\bar{x} = (x_1, \dots, x_n)$ and a tuple of values $\bar{v} = (v_1, \dots, v_n)$ such that for all $1 \leq i \leq n$ we have that $x_i \in X$ and $v_i \in \mathbb{R}_+$, we define the valuation $vl[\bar{x}/\bar{v}]$ such that for all $x \in X$:

$$vl[\bar{x}/\bar{v}](x) = \begin{cases} v_i & \text{if } \exists 1 \leq i \leq n : x = x_i \\ vl(x) & \text{otherwise} \end{cases}$$

Given a valuation $vl \in \mathcal{V}_X$ and a transformation function for a variable $var \in X$, $f_{var} \in \mathcal{TV}$, we define the valuation $f_{var}[vl]$ for all $x \in X$ as:

$$f_{var}[vl](x) = \begin{cases} f_{var}(vl(x)) & \text{if } x = var \\ vl(x) & \text{otherwise} \end{cases}$$

The satisfaction of a fuzzy constraint C by a valuation $vl \in \mathcal{V}_X$ considering a t -norm Δ , denoted by $vl \models C$, is defined as follows:

$$\begin{aligned} vl &\models True \\ vl &\models fr(x_1, \dots, x_n) \text{ iff } fr(vl(x_1), \dots, vl(x_n)) > 0 \\ vl &\models F_1, F_2 \text{ iff } (vl \models F_1 \vee vl \models F_2) \wedge \Delta(F_1, F_2) > 0 \end{aligned}$$

Next, we introduce the notion of an extended finite automaton. Essentially, this is a finite automaton extended with a set of variables; these variables will be used to establish constraints for triggering transitions.

Definition 6 A fuzzy automaton is a tuple $A = (S, Acts, X, X_0, s_0, T)$ where:

- S : finite set of states.
- Acts: set of actions, divided into a set of inputs I and a set of outputs O . An input action consists of a name and empty tuple of variables. An output action is composed of a name and maybe a tuple of expressions over the set of variables.
- X : a set of variables.
- $X_0 : X \rightarrow \mathbb{R}_+$ a function that assigns the initial values to the variables.

- $s_0 \in S$: the initial state.
- $T \subseteq S \times \text{Acts} \times \mathcal{C}_X \times \mathcal{P}(\mathcal{TV}) \times S$: a set of transitions satisfying next conditions:
 - Output transitions have associated True as constraint.
 - Input transitions may have associated a fuzzy constraint whose free variables appear in the input action.
 - The set of transformation functions associated with output transitions is the empty set.
 - Given an output transition outgoing from a state s there does not exist an input transition outgoing from s .

A configuration of A is a pair (s, vl) where $s \in S$ is the current state and $vl \in \mathcal{V}_X$ is the valuation corresponding to the current value of the variables belonging to X .

Given a configuration (s, vl) , a tuple of variables $\bar{x} = (x_1, \dots, x_n)$ belonging to X and a tuple of real values $\bar{v} = (v_1, \dots, v_n)$, if an input action $a(\bar{v})$ is received then a transition $(s, a(\bar{x}), C, \{f_{x_1}, \dots, f_{x_n}\}, s')$ can be fired if $vl[\bar{x}/\bar{v}] \models C$. In this case, the configuration will change to $(s', f_{x_1}(\dots f_{x_n}(vl[\bar{x}/\bar{v}]) \dots))$. If a transition $(s, a(e_1, \dots, e_n, True), s')$ is triggered, the output a is produced for the expressions e_1, \dots, e_n that are evaluated considering the valuation vl . Then the configuration will change to (s', vl) .

A FZA is a directed graph which transitions are labeled by an action and a fuzzy constraints. The input actions consist of a name (preceded by ?) and a tuple of variable names. The outputs also have a name (preceded by !), followed by a tuple of expressions. A transition may also have a variable transformation associated with it. A variable transformation assigns to each variable a real value. Intuitively, when the automaton receives an input action, the fuzzy constraints associated to the transitions labeled by this input are evaluated with the variable values mapped by the input action. If the grade of confidence is higher than 0, then the transitions can be performed and the variable transformations may be applied. If the evaluations of several fuzzy constraints are higher than 0, then the transition with the highest grade of confidence will be performed. When an output action is produced, the associated expressions are evaluated considering the current value of the variables.

Example 1. Let us consider the FZA depicted in Figure 3. This is a simplification of a FZA that can send out alerts regarding the risk of a patient to suffer diabetes. The complete version of the automaton takes into account the gender and age of the patients, as well as other aspects that can affect normal blood sugar level ranges. However, for the sake of clarity and simplicity, in this example, we will only analyze the blood sugar level measures with respect to the normal range corresponding to a healthy adult before meals, that is, $[90, 100]$. In order to produce a diagnosis, the blood sugar levels of the patients will be checked.

The automaton presents 4 states and 7 transitions, being s_0 the initial state. After the identifier of the patient is read and stored in variable a ($?id(a)$), the automaton checks each measure and determines if they fall within the expected range. The input action $?glucose(gl, t)$ stores the measure and the time at which it was taken in variables gl and t , respectively. Depending on the value of gl two different transitions can be triggered. Let us assume $gl = 102$. In this case, the measure does not belong to the normal blood

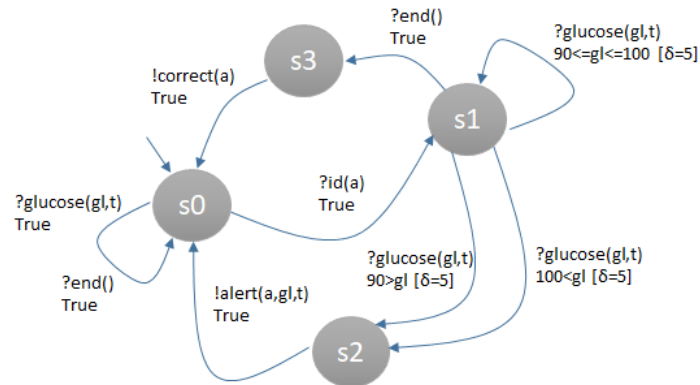


Figure 3: FZA example

sugar level range. However, a small deviation is allowed ($\delta = 5$) and therefore, the evaluation of the fuzzy constraint associated to one of the transitions outgoing from state $s1$, $90 \leq gl \leq 100$ [$\delta = 5$], provides a positive level of confidence, specifically, 0.6. It means that, with probability 0.6, the value of gl can be considered *acceptable*. The rest of the measures will be processed until either all the data of the patient have been read or there exists a positive confidence that a value is out of the *acceptable* range. In the case that $gl = 120$, the transition labeled with the fuzzy constraint $90 > gl \parallel 100 < gl$ [$\delta = 5$] will be triggered with a level of confidence 1. Then, it will produce an output action, $!alert(a, gt, t)$, that informs about the possibility that the patient suffers the risk of having a diabetes disease.

3 Our Approach in a Nutshell

This section summarizes our model-driven approach for modeling FZA and transforming them into code for data analysis. This approach is illustrated in Figure 4 and detailed as follows:

1. **FZA modeling.** The domain expert must graphically design the FZA for a specific protocol, such as diagnosis support for a specific disease. The designed model must conform to our metamodel described in Section 4.1.1.
2. **FZA validation.** Once the model has been designed, the editor validates it by means of the metamodel constraints and the validation rules defined in Section 4.1.2. In the case the model does not comply any of these conditions, the editor shows the errors which must be solved.
3. **Model-to-code transformation.** The model is automatically transformed into the AUNTY code as explained in Section 4.2.1.
4. **Code import.** The generated code is imported by the AUNTY tool with the goal of being used for data analysis.
5. **Data analysis.** At this point, the AUNTY tool can analyse data on the basis of the restrictions considered in the designed model.

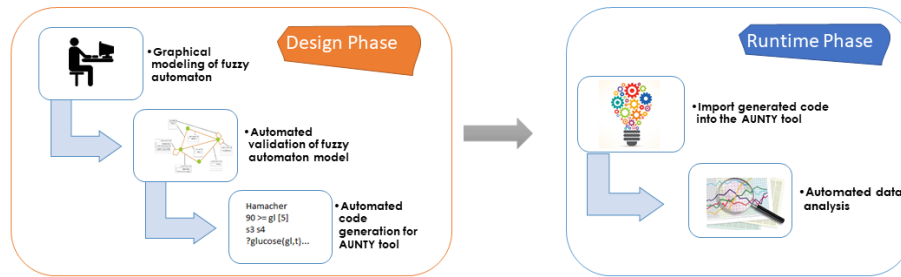


Figure 4: Phases of our model-driven approach

The phases 1-3 are performed at design time while steps 4-5 correspond to runtime execution.

We would like to highlight that the main aim of our model-driven approach is to facilitate the design of FZA through the development of a domain-independent editor that allows non-technical users to graphically define FZA in a easy way, avoiding they have to write code that requires technical skills. To reach this goal, we have followed the following steps:

- **Definition of the FZA metamodel.** A metamodel for FZA has been proposed. This is domain-independent, and therefore can be used for designing models that require the use of the key features of this formalism, that is, the specification of uncertainty and imprecision in the analysis of the data.
- **Development of a graphical modeling editor.** An editor for the FZA metamodel has been developed with the aim of facilitating non-technical experts the design of the protocols to be applied for the analysis of critical information, based on their expertise knowledge.
- **Model validation process.** In addition to the constraints imposed by the metamodel, it is necessary to check that it satisfies the validation rules that allow us to determine that the model is well-formed.
- **Model transformation process.** Once a correct model has been designed, we propose a model transformation process in order to automatically transform such a model into code that can be used by the AUNTY tool for analyzing data.

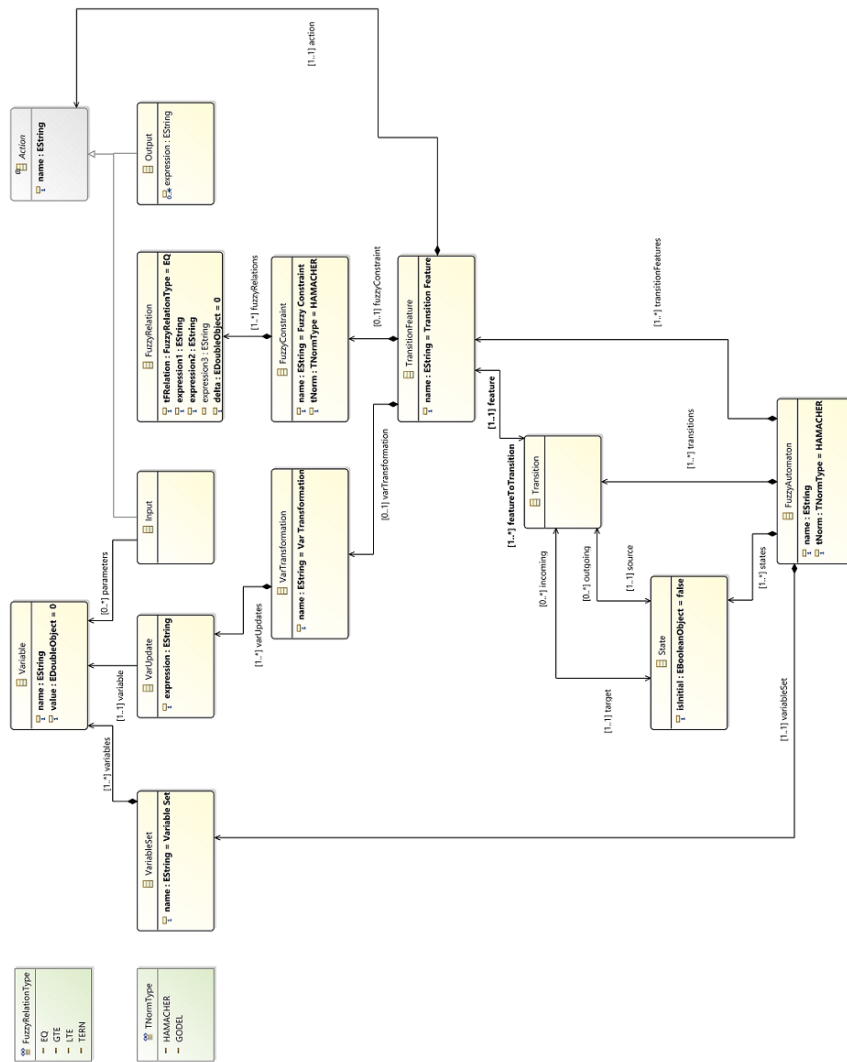


Figure 5: FZA Metamodel

4 Domain-specific Modeling Language for Fuzzy Automata

In this section, we propose a graphical DSML that will allow users to define FZA. Following, we introduce the components that define it: (a) the *abstract syntax* consisting of both a metamodel—a model describing FZA concepts and the relationships between them—and a collection of validation rules to determine whether a model is well formed—; (b) the *concrete syntax* that consists of the graphical symbols that will be used for designing models; and (c) the model-to-text transformation for software automatic generation.

4.1 Abstract Syntax

The abstract syntax of a DSML is composed of a metamodel, in which the language concepts and relationships between them are defined, as well as the conditions over elements and relationships that the models must comply with in order to ensure they are well-formed.

4.1.1 Fuzzy Automaton Metamodel

Figure 5 depicts the proposed metamodel for defining FZA. Next, we describe the metaclasses and the relationships of which is composed it:

- *FuzzyAutomaton*: The main metaclass of the metamodel. The root of a FZA is an instance of *FuzzyAutomaton*, which must contain states (*State*), a set of variables (*VariableSet*), transitions (*Transition*) and transition elements (*TransitionFeature*). Every FZA is identified by a unique name and has associated the T-norm that will be used to combine the confidence values obtained from the evaluation of fuzzy constraints associated to the executed transitions. The *tNorm* attribute can take the *GODEL* and *HAMACHER* values.
- *State*: A state can be either an initial or not initial state.
- *Variable*: An identifier whose value might change during the execution of the automaton. The value corresponds to double type.
- *VariableSet*: A named collection of variables (*Variable*) that may be used as input parameters of input actions (*Input*) and may be included in expressions associated to output actions (*Output*), fuzzy relations (*FuzzyRelation*) or variable updates (*VarUpdate*).
- *Transition*: A transition connects two states (*State*), a source state and a target state. It also has associated some features (*TransitionFeature*) that corresponds to the *FuzzyConstraint*, *Action* and *VarTransformation* related to the execution of the transition.
- *TransitionFeature*: A container element that consist of, at least, an *Action* but can also contain a *FuzzyConstraint* and/or a *VarTransformation*.
- *Action*: An action can be either an input action (*Input*) or an output action (*Output*).
- *Input*: An input action corresponds to an stimuli of the environment and can have associated a set of variables (*Variable*) that represents input parameters.

- *Output*: An output action produces information that corresponds to the evaluation of the associated expression.
- *FuzzyConstraint*: A fuzzy constraint, identified by a name corresponds to a set of fuzzy relations (*FuzzyRelation*). The *tnorm* attribute establishes how the values obtained from the evaluation of the fuzzy relations have to be combined.
- *FuzzyRelation*: A relation whose type can be *EQ*, *GTE*, *LTE*, corresponding to the binary operators =, >= and <=, respectively, or the ternary relation *TERN* that represents an interval. Depending on the type of the relation, two or three expressions have to be assigned to it.
- *VarTransformation*: A named group of variable updates (*VarUpdate*).
- *VarUpdate*: An expression associated to a variable (*Variable*) that corresponds to the update of its value.

4.1.2 Validation Rules for FA Metamodel

In addition to the metamodel, the abstract syntax requires the definition of validation rules. A model conforms a metamodel if it satisfies the restrictions imposed by the metaclasses and the relations between them and comply the validation rules. Table 1 presents the validation rules defined in our DSML.

Metaclass	Validation rules
<i>State</i>	A FZA must have an initial state. A FZA has only one initial state. A state has at least an incoming or an outgoing transition.
<i>Variable</i>	The first character of the identifier of a variable (<i>name</i>) cannot be a digit. The name of a variable must be unique.
<i>Output</i>	Expressions associated with outputs only contain arithmetic operators. Expressions are type-correct.
<i>Fuzzy Relation</i>	Expressions associated with fuzzy relations only contain arithmetic operators. Expressions only contain variables defined in the <i>VariableSet</i> . Expressions are type-correct. The attribute <i>expression3</i> is mandatory for ternary fuzzy relation, i.e., those whose <i>type</i> is <i>TERN</i> . The attribute <i>expression3</i> cannot have a value in binary fuzzy relations, i.e., those whose <i>type</i> is <i>EQ</i> , <i>GTE</i> or <i>LTE</i> .
<i>Var Update</i>	The expressions associated with variable updates only contain arithmetic operators. Expressions are type-correct.

Table 1: Validation rules for fuzzy automaton metamodel.

4.2 Concrete Syntax

The concrete syntax of a DSML allows us to establish a relationship between metamodel concepts and their textual or graphical representation. In our case, we have defined a graphical concrete syntax that provides a graphical notation for each of the elements that can be used in the design of a FZA model. Figure 6 shows the concrete syntax for FZA models.

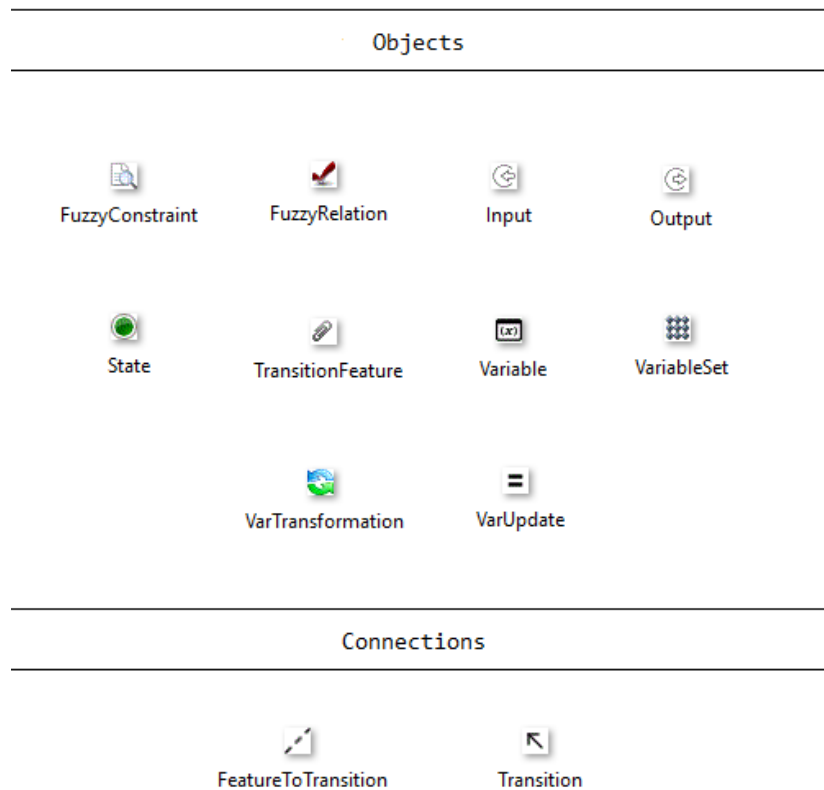


Figure 6: Graphical concrete syntax

4.2.1 Model-to-text Transformation

The last component of our DSML corresponds to the transformation of FZA models into AUNTY code. The transformation will allow users to analyze data against the model of interest. Figure 7 shows the model transformation integrated in our framework. In order to provide the automatic model-to-code transformation, we have created the *FZAtoAUNTYcode* module, which has been implemented in Eclipse Epsilon Generation Language (EGL) [Kolovos et al., 2020].

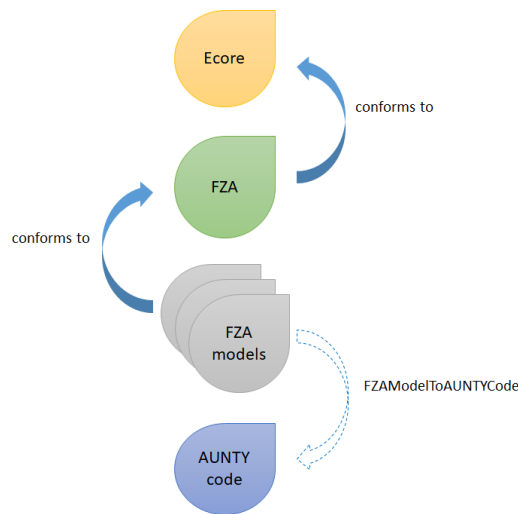


Figure 7: FZA-AUNTY code transformation

5 Graphical Editor

In order to give support to users for designing FZA models, we have developed a graphical modeling editor¹. We have used the Epsilon family of languages [Kolovos et al., 2020] to implement it. Epsilon provides EuGENia, a tool which automatically generates the models needed for the implementation of a Graphical Modeling Framework (GMF) editor resulting from a single annotated Ecore metamodel. This editor allows to create FZA models conforming to the FZA metamodel which will be stored as XML Metadata Interchange(XMI) files. The editor is composed of the tool palette, the canvas and the property view.

The palette has been customized using Epsilon Object Language (EOL), an imperative language inspired by OCL. The elements in the palette have been classified in two groups:

- *Objects*. It includes the tools corresponding to the following metamodel classes: *FuzzyConstraint*, *FuzzyRelation*, *Input*, *Output*, *State*, *TransitionFeature*, *Variable*, *VariableSet*, *VarTransformation*, *VarUpdate*.
- *Connections*. It contains the *Transition* tool for connecting states (*State*), and the *FeatureToTransition* tool for linking transitions (*Transition*) to transition features (*TransitionFeature*).

The canvas is the area where the elements in the palette can be placed, in a drag-and-drop fashion, to define models that conform to the FZA metamodel. The attributes of the elements can be established both in a graphical way and using the properties view. Notice that a particular palette tool can be used in a specific component if the obtained model conforms to our metamodel and satisfy the validation rules defined in Section 4.1. Figure 8 shows a screenshot of the graphical editor.

¹ MODELIFY: <https://github.com/mgmerayo/MODELFY>

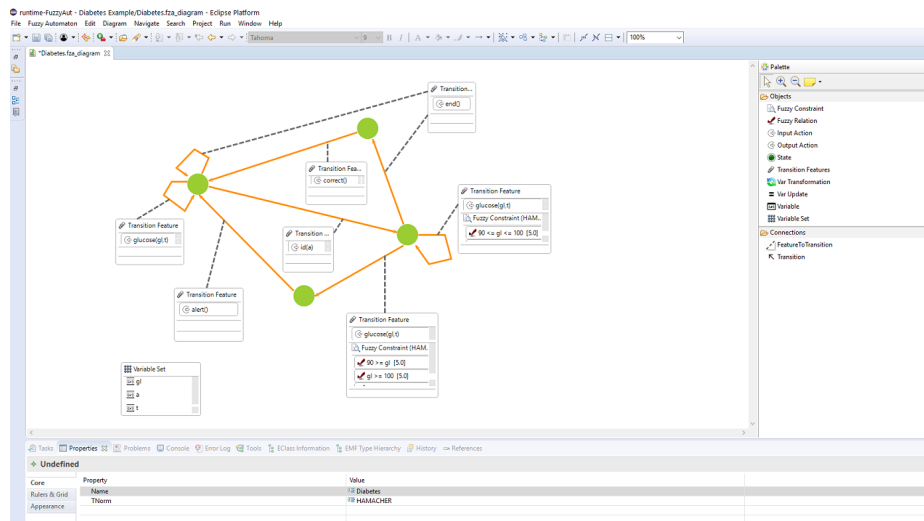


Figure 8: FZA graphical editor

6 Case Study: Early Detection of Heart Diseases

The goal of this case study is to show how our proposal (see Section 3) can be applied for the design of a real protocol that requires the incorporation of uncertainty and imprecision.

AUNTY tool provides users with a functionality that allows to load fuzzy automata from a text file with a specific format, and saved it for being used in the analysis of data. Data must be given in the form of sequences of input actions corresponding to the considered automaton. AUNTY is able to apply the sequence of input actions to the automaton and provides the produced outputs together with the degree of confidence associated with them. In this case study, we focus on prediction of heart problems. Specifically, we have modeled the behavior of the heart based on the data that can be extracted from ECGs (electrocardiograms): heartbeats per minute and the time elapsed between two successive R-waves of the QRS signal on the electrocardiogram (RR interval). This model, corresponding to a fuzzy automaton, has been designed using our graphical editor. The automatic model-to-text transformation integrated in this editor has provided us with the file corresponding to the code required by the AUNTY tool for using it in the analysis of the ECGs. Our model considers normal levels of these parameters [Rijnbeek et al., 2014, Haarmark et al., 2010]. The application of the data of the patients collected from ECGs to our model will allow us to alert about the level of risk of a patient to suffer a heart problem.

Next, we will apply the steps described in our model-driven approach for modeling the behavior of the heart by means of a FZA and transforming it into code for data analysis.

In particular, the data that the automaton must manage corresponds to the next parameters:

- *Gender*: Man and Woman.
- *Age*: Eight groups of age have been defined: [20, 29], [30, 39], [40, 49], [50, 59],

[60, 69], [70, 79], [80, 89] and [90, 99].

- *Heartbeats*. The range of heartbeats per minute (bpm) for healthy patients according to their gender and age.
- *RR waves*. The range of RR waves duration (measured in milliseconds) for healthy patients, according to their gender, age and bpm.

In the case that an *abnormal* data is detected, an alarm, associated with a grade of confidence, will be sent out.

The automaton has 130 states and 258 transitions. There are two transitions outgoing from the initial state corresponding to the gender of the patient. After that, each of the reached states has 8 transitions, one per age group. Each of these age groups has associated a *sub-automaton* with 7 states and a transition to the initial state that indicates that all the data for a patient have been processed. Each minute, the automaton checks if the number of beats falls within the normal amount of bpm in the age range. If it does, then the automaton does not take into account the RR intervals in that minute and marks it to be ok. Otherwise, the automaton processes each RR interval. If there is at least one interval out of range, an alarm is raised (with a certain grade of confidence).

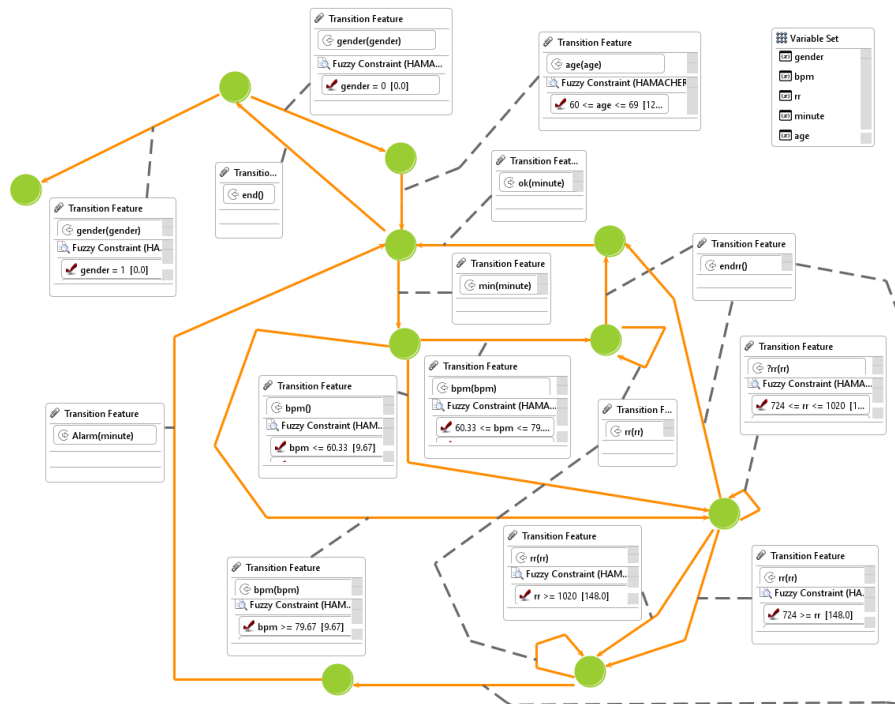


Figure 9: FZA for detecting heart problems protocol

Figure 9 shows a partial view of the FZA designed with the graphical editor. Additionally, Listing 1 in Appendix A presents a fragment of the file corresponding to the

code required by the AUNTY tool for the analysis of data generated from the graphical version. The file can be found at <https://github.com/mgmerayo/MODELFY>.

As we mentioned previously, the ranges and deviations used to define the fuzzy constraints associated to the transitions have been taken from [Rijnbeek et al., 2014]. In the case of the age, the δ value is obtained from the 20% of the highest value of each age range. The idea is that the patients could be wrongly classified according to their age. For instance, if a patient is 53 years old then we should classify him/her in the group that considers the range [50,59]. However, it may happen that the patient has a very healthy life style and, therefore, his/her heart is *younger*. Hence, we should also classify this patient in the age group corresponding to the range [40, 49] and decide in which group fits better the collected. Regarding the heartbeats, we had applied the estimations made in [Hozo et al., 2005]: if the size of a sample exceeds 25 then the median itself is the best estimator for the mean and the best estimator for the standard deviation is $\sigma \approx b - a/6$ where a is the smallest value of the sample (the 2nd percentile in our case) and b is the largest value of the sample (the 98th percentile in our case).

The database that we use [Rijnbeek et al., 2014] has data from 13354 patients and, therefore, we can base our limits on the median of each range while δ is based on the standard deviations of each range. Concerning RR waves, we have used the data from [Haarmark et al., 2010]. The problem in this case was that we only had the information of the RR waves duration for the patients of the age range [30–39]. So, if we had used these limits for all the patients, then the prediction would have been erroneous. Therefore, we also considered another work [11] where the duration of the RR waves is derived from data corresponding to heartbeats. So, our limits are based on the application of the following formula $RR_{ms} \approx 60000/bpm$ to the bpm data obtained from [Rijnbeek et al., 2014]. Once we have designed the FZA using our graphical editor, it has been automatically transformed to the code that the AUNTY tool use for execution. With the goal of assessing the usefulness of the FZA, we have considered the MIT/BIH Arrhythmia Database Directory [Moody and Mark, 2001]². This study includes data from 48 ECGs with a duration of 30 min. All of them present some heart pathology. Specifically, 48% of the samples have been annotated in the database as representative cases of routine clinical recordings while the remaining 52% reflect uncommon cases of arrhythmia.

Among the results that we obtained from the analysis of the data we have selected the results corresponding to five different patients as example: 2 males and 3 females. For each patient, our model has taken three age ranges into consideration. The first range corresponds to the one to which the patient *actually* belongs. The other two ranges are the adjacent ones. We have applied our automaton to the record of each patient and we have recorded the signals that our model produces per minute during the analysis of the data. Table 2 shows the results obtained when the *real range* of the patients is considered. It only presents the data corresponding to the first 30 minutes of the process. Each column shows the *grade of confidence* associated to the *alert* and *ok* signals obtained per minute.

Next, we briefly explain some of the inferences we can draw from the results obtained from the evaluation of the selected patients. In order to simplify the analysis, we base on the graphical representation of the results. We use 3 graphics per patient corresponding to the results related to the 3 ranges of age considered.

Patient 221 is a 83 year-old male. The analysis of his record (see Figure 10) shows that he presents some kind of arrhythmia. Even if we consider the adjacent ranges of age.

Patient 107 is a 63 year-old male. We can see in Figure 11 that the results show intermittent periods of time where we have no solid evidence of any heart issue, followed

² <https://physionet.org/physiobank/database/mitdb/>

min	Patient 221		Patient 107		Patient 220		Patient 207		Patient 222	
	ok	alert	ok	alert	ok	alert	ok	alert	ok	alert
1	0.0	1.000000	0.000000	0.600000	0.000000	0.000000	0.000000	1.000000	0.000000	0.285714
2	0.0	1.000000	0.796187	0.500000	1.000000	0.085925	0.000000	0.095238	0.000000	0.380952
3	0.0	1.000000	1.000000	0.347799	1.000000	0.144411	0.000000	0.761905	1.000000	0.000000
4	0.0	1.000000	1.000000	0.426471	1.000000	0.147127	0.000000	1.000000	0.950450	0.000000
5	0.0	1.000000	0.483892	0.400000	1.000000	0.381852	0.000000	1.000000	0.950450	0.000000
6	0.0	1.000000	1.000000	0.421013	0.000000	0.190476	0.000000	1.000000	0.866157	0.000000
7	0.0	1.000000	1.000000	0.351515	0.000000	1.000000	0.000000	0.380952	0.770284	0.095238
8	0.0	1.000000	1.000000	0.421013	0.000000	0.285714	0.000000	0.761905	0.987988	0.190476
9	0.0	1.000000	0.779750	0.500000	1.000000	0.428782	0.000000	0.666667	0.000000	0.380952
10	0.0	1.000000	0.138725	0.400000	0.531410	0.761905	1.000000	0.092575	0.000000	1.000000
11	0.0	1.000000	0.319527	0.500000	0.000000	0.285714	1.000000	0.089940	0.000000	1.000000
12	0.0	1.000000	1.000000	0.343994	1.000000	0.846245	1.000000	0.253190	0.000000	1.000000
13	0.0	1.000000	0.155161	0.500000	1.000000	0.596710	1.000000	0.432268	0.000000	1.000000
14	0.0	1.000000	0.347536	0.500000	0.000000	0.666667	0.897462	0.952381	0.000000	1.000000
15	0.0	1.000000	0.165531	0.600000	0.000000	0.761905	1.000000	0.897748	0.000000	1.000000
16	0.0	1.000000	0.184776	0.500000	0.000000	0.857143	0.938752	1.000000	0.000000	1.000000
17	0.0	1.000000	0.081504	0.500000	0.000000	0.857143	0.798727	1.000000	0.000000	1.000000
18	0.0	1.000000	0.244470	0.500000	0.952381	0.996246	0.921689	1.000000	0.000000	1.000000
19	0.0	1.000000	0.153048	0.600000	0.000000	0.380952	0.892913	1.000000	0.000000	0.190476
20	0.0	1.000000	0.133759	0.600000	0.000000	0.761905	0.831243	1.000000	0.000000	0.476190
21	0.0	0.818182	0.000000	0.700000	0.000000	0.000000	0.693737	1.000000	0.000000	1.000000
22	0.0	0.909091	0.894806	0.500000	0.000000	0.571429	0.818086	1.000000	0.000000	1.000000
23	0.0	0.636364	1.000000	0.358694	0.000000	0.666667	0.611593	1.000000	0.000000	1.000000
24	0.0	0.272727	0.000000	0.700000	0.000000	0.380952	0.557746	1.000000	0.000000	1.000000
25	0.0	0.818182	1.000000	0.365553	0.000000	0.095238	0.699785	1.000000	0.000000	1.000000
26	0.0	0.636364	0.108683	0.600000	0.000000	0.285714	0.000000	1.000000	0.000000	1.000000
27	0.0	0.818182	0.074373	0.600000	1.000000	0.193868	0.000000	1.000000	0.000000	1.000000
28	0.0	1.000000	1.000000	0.347799	1.000000	0.421764	0.000000	1.000000	0.000000	1.000000
29	0.0	0.636364	0.102818	0.600000	0.000000	0.380952	0.000000	1.000000	0.000000	1.000000
30	0.0	0.727273	0.000000	0.600000	0.000000	0.476190	0.000000	1.000000	0.000000	1.000000

Table 2: Grades of confidence per patient and minute for real range.

by periods where evidence of a possible arrhythmia can be appreciated. It is worth noticing that peaks related to the confidence of the patient being healthy correlate with valleys corresponding to the confidence of the patient having heart issues.

Patient 220 is a 87 year-old female. In this case (see Figure 12), the results are really interesting since, overall, we can observe that the first half of the results matches with the expected heart parameters of a person whose age is between 80 and 90 years old. However, the second half of the record tends to fit better with the usual parameters of a person older than 90 years old. This may indicate that the heart tissue of the patient has some damage that makes the behavior of the heart corresponds to the one of an older person.

Patient 207 is a 89 year-old female. The analysis of the data (see Figure 13), corresponding to the first minutes, exhibits that of the patient has some risk of having a heart problem. However, around minute 10 it experiments a noticeable improvement. This may be caused by a treatment, since after a few minutes we can observe that the degree of confidence associated to the presence of a possible heart problem increases while the level of confidence related to a healthy heart decreases.

Patient 222 is a 84 year-old female. In this case (see Figure 14), we can observe that during the first minutes, the data fit in the expected behavior of a healthy heart. However,

after minute 10, alerts maybe related to an abnormal behavior are raised again.

As a result, we can conclude that the representation of the protocol by means of a FZA has appropriately captured the protocol for detecting potential heart problems.

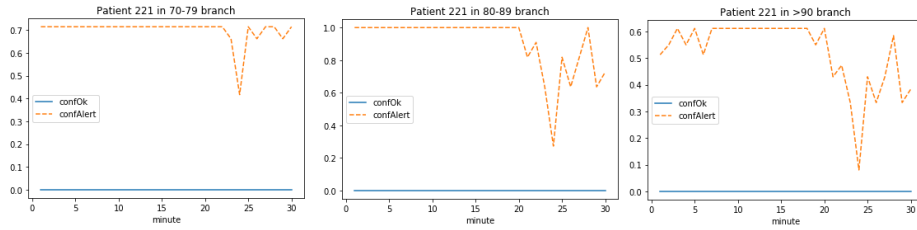


Figure 10: Patient 221

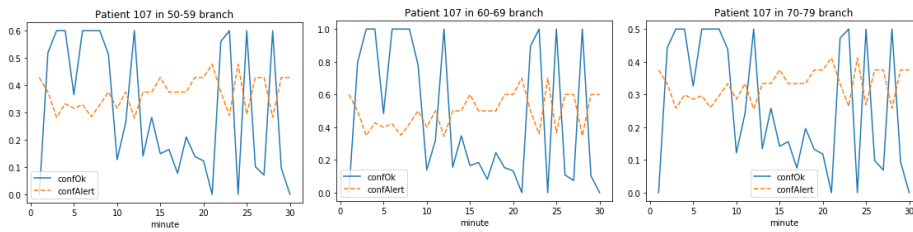


Figure 11: Patient 107

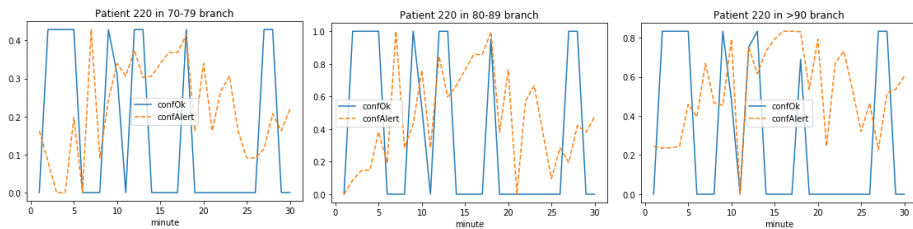


Figure 12: Patient 220

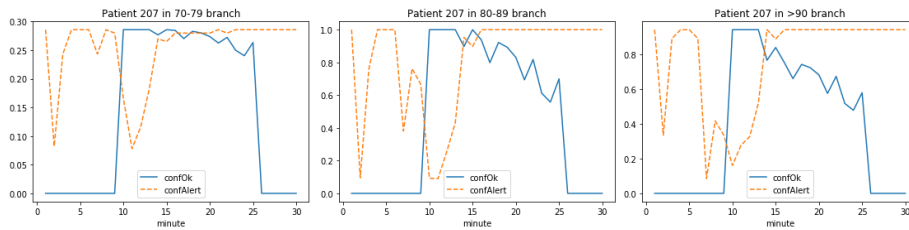


Figure 13: Patient 207

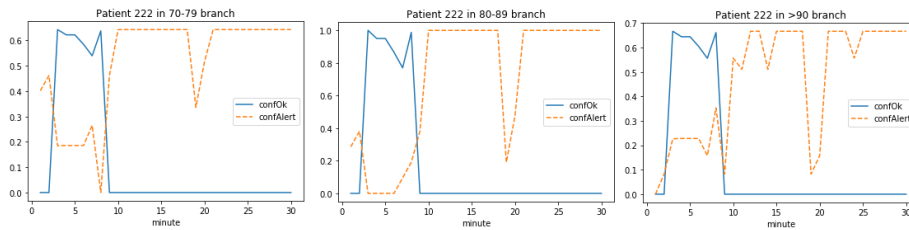


Figure 14: Patient 222

7 Threats To Validity

The results obtained from our experiments have possible threats to their validity. In this section we present them. Concerning threats to internal validity, which consider uncontrolled factors that might be responsible for the obtained results, the main threat is associated with the possible faults introduced in the development of the experiments, since the results could be compromised. In order to reduce the impact of this threat we tested our graphical editor. We focused in checking the correct behavior of the validation rules and the model to text transformation. We did it applying mutation testing techniques with the goal of injecting faults of interest in different models. It helped us to determine the capacity of the editor for detecting these error. In addition, we have replicated the performed experiments several times to check the robustness of the results.

The main threat to external validity, which concerns conditions that allow us to generalize our findings to other scenarios, the most important one corresponds to the model we used to perform the experiments. Such a threat cannot be addressed since we had to focus on a specific model. Nevertheless, any other model could have been used.

We have also considered threats to construct validity, which are related to the reality of our experiments, that is, whether they reflect real-world situations or not. In our work, the main construct threat is what would happen if we used our editor for designing much more complex models. Despite being an important threat, our editor is able to support large models. Therefore, we consider that the threat is not as disturbing.

Modeling	[Gomes and Oliveira, 2009, Leemans and Amalio, 2012] [Masci et al., 2013, Kwiatkowska et al., 2014, Singh et al., 2015] [Jiang et al., 2010, Méry and Singh, 2012, Chen et al., 2014] [Gomes and Butterfield, 2016, Hoang et al., 2016] [Costa et al., 2019, Wang et al., 2020]
Verification	[Alur et al., 2004, Jee et al., 2010, Jiang et al., 2010] [Masci et al., 2013, Kwiatkowska et al., 2014, Chen et al., 2014] [Singh et al., 2015, Arcaini et al., 2016] [Gomes and Butterfield, 2016, Khan et al., 2018]
Validation	[Arney et al., 2007, Jee et al., 2010, Arcaini et al., 2016] [Hoang et al., 2016, Ai et al., 2020]
Code Generation	[Gomes and Oliveira, 2009, Jiang et al., 2010]

Table 3: Summary of formal proposals by development phase

8 Related Work

In the field of decision making, there exist many proposals that help decision makers to select among different alternatives for solving a problem [Cabrerizo et al., 2010, Herrera-Viedma et al., 2014, Loewer and Laddaga, 1985, Perez et al., 2016, Pérez et al., 2018, Xu et al., 2017, Wu and Xu, 2012], some of them supported by graphical tools [Palomares et al., 2014, Ureña et al., 2015]. Nevertheless, they mainly focus on situations in which several experts need to achieve a common solution to a decision problem, while our solution aims at providing experts in a particular domain with a tool that allows them to design rigorously protocols that assist in decision-making processes such as diagnosis.

In the literature, we can find works that propose the use of formal methodologies for the development of medical devices and systems. Most of the proposals are applied to pacemakers [Gomes and Oliveira, 2009, Jee et al., 2010, Jiang et al., 2010, Méry and Singh, 2012, Leemans and Amalio, 2012, Kwiatkowska et al., 2014, Ai et al., 2020, Wang et al., 2020], infusion pumps [Alur et al., 2004, Arney et al., 2007, Masci et al., 2013, Singh et al., 2015, Costa et al., 2019] and hemodialysis systems [Arcaini et al., 2016, Gomes and Butterfield, 2016, Hoang et al., 2016, Khan et al., 2018].

These proposals address different phases of the development process, highlighting the modeling of systems, application of techniques of verification and validation. To a lesser extent they are also applied to code generation. The most used formalisms in these proposals are, mainly, Timed Automata and others based on states such as Hybrid Input Output Automata, Extended Finite State Machines, Petri Nets, Abstract State Machines, Z, Algebraic State Transition Diagrams or EVENTB modelling. The classification of the publications based on both the development phase to which they are applied and the formalism they use appear in Tables 4 and 3, respectively.

Another aspect of medicine in which formal methods have also been applied is the design of medical protocols [ten Teije et al., 2006, Mordecai, 2019], which facilitate and allow specialists a better understanding of the process of attention to the patients, improving their security.

Nevertheless, these approaches do not take into account uncertainty and imprecision, something critical when dealing with medical systems. As it has been shown along this paper, fuzzy logic allows us to overcome this drawback. To the best of our knowledge, there does not exist a graphical DSML for FZA that allows to design protocols for giving

Timed Automata	[Jee et al., 2010, Jiang et al., 2010] [Pajic et al., 2012, Ai et al., 2020]
Hybrid Input Output Automata	[Chen et al., 2014, Wang et al., 2020]
Extended Finite State Machine	[Alur et al., 2004, Arney et al., 2007]
Petri Nets	[Costa et al., 2019]
Abstract State Machine	[Arcaini et al., 2016]
Z	[Gomes and Oliveira, 2009]
Algebraic State Transition Diagrams	[Fayolle et al., 2016]
EVENTB modelling	[Méry and Singh, 2012]

Table 4: Summary of proposals by formalism

support to specialists in the diagnosis of diseases.

9 Conclusions and Future Work

MODELFY is a model-driven solution for designing a decision-making process based on fuzzy information given by means of a FZA. The solution is composed of a DSML together with a graphical tool that allows experts that have a high knowledge in a particular domain, but do not have the necessary technical knowledge, to model FZAs in an intuitive way. The designed models can be then automatically transformed into code which can be executed on the AUNTY tool for fuzzy data analysis.

To demonstrate its feasibility, this solution has been applied to a non-trivial diagnostic medical protocol for detecting potential heart problems. The results have confirmed that the graphical modeling tool is useful for defining such a protocol in a user-friendly and rigorous manner, hiding all implementation details from domain experts.

Therefore, MODELFY is a novel solution for bringing FZA formalism closer to domain experts, who possess the appropriate knowledge for conducting the decision-making process. Our case study allows us to state that our solution is useful for defining in a precise way the protocols that guide the analysis of fuzzy data and automating their application. This solution can be used to model any decision problem that can be represented by means of a fuzzy automaton.

As a future work, we plan to extend our FZA DSLM and editor with different data types and t-norms to give support for the modeling and analysis of other application domains. Thanks to the use of model-driven techniques, FZA graphical models only have to be designed once by the domain experts. By creating and adding new model-to-code transformation rules, these graphical models could be automatically implemented in the programming languages required for performing fuzzy data analysis by means of the appropriate tools.

Acknowledgements

This work has been supported by the Spanish MINECO/FEDER project FAME (RTI2018-093608-B-C31 and RTI2018-093608-B-C33); the Comunidad de Madrid project FORTE-CM (S2018/TCS-4314) co-funded by EIE Funds of the European Union.

References

- [Ai et al., 2020] Ai, W., Patel, N. D., Roop, P. S., Malik, A., Trew, M. L.: ‘Closing the Loop: Validation of Implantable Cardiac Devices With Computational Heart Models’; *IEEE Journal of Biomedical and Health Informatics*, 24, 6 (2020), 1579–1588. <https://doi.org/10.1109/JBHI.2019.2947007>
- [Alur et al., 2004] Alur, R., Arney, D., Gunter, E. L., Lee, I., Lee, J., Nam, W., et al.: ‘Formal specifications and analysis of the computer-assisted resuscitation algorithm (CARA) Infusion Pump Control System’; *International Journal on Software Tools for Technology Transfer*, 5, 4 (2004), 308–319. <https://doi.org/10.1007/s10009-003-0132-7>
- [Arcaini et al., 2016] Arcaini, P., Bonfanti, S., Gargantini, A., Riccobene, E.: ‘How to Assure Correctness and Safety of Medical Software: The Hemodialysis Machine Case Study’; In M. Butler, K.-D. Schewe, A. Mashkooor & M. Biro (Eds.), *Abstract State Machines, Alloy, B, TLA, VDM, and Z*. Cham: Springer International Publishing (2016), 344–359. https://doi.org/10.1007/978-3-319-33600-8_30
- [Arney et al., 2007] Arney, D., Jetley, R., Jones, P., Lee, I., Sokolsky, O.: ‘Formal Methods Based Development of a PCA Infusion Pump Reference Model: Generic Infusion Pump (GIP) Project’; In *2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007)* (2007), 23–33. <https://doi.org/10.1109/HCMDSS-MDPnP.2007.36>
- [Brambilla et al., 2017] Brambilla, M., Cabot, J., Wimmer, M.: ‘Model-Driven Software Engineering in Practice: Second Edition’; (2nd ed.). Morgan & Claypool Publishers (2017).
- [Borgwardt et al., 2017] Borgwardt, S., Cerami, M., Peñaloza, R.: ‘The complexity of fuzzy EL under the Łukasiewicz T-norm’; *International Journal of Approximate Reasoning*, 91 (2017), 179–201. <https://doi.org/10.1016/j.ijar.2017.09.005>
- [Boubeta-Puig et al., 2015] Boubeta-Puig, J., Ortiz, G., Medina-Bulo, I.: ‘Medit4CEP: A model-driven solution for real-time decision making in SOA 2.0’; *Knowledge-Based Systems*, 89 (2015), 97–112. <https://doi.org/10.1016/j.knosys.2015.06.021>
- [Boubeta-Puig et al., 2019] Boubeta-Puig, J., Díaz, G., Macià, H., Valero, V., Ortiz, G.: ‘Medit4CEP-CPN: An approach for complex event processing modeling by prioritized colored petri nets’; *Information Systems*, 81 (2019), 267–289. <https://doi.org/10.1016/j.is.2017.11.005>
- [Cabrerizo et al., 2010] Cabrerizo, F. J., Heradio, R., Pérez, I. J., Herrera-Viedma, E.: ‘A Selection Process Based on Additive Consistency to Deal with Incomplete Fuzzy Linguistic Information’; *Journal of Universal Computer Science*, 16, 1 (2010), 62–81. <https://doi.org/10.3217/JUCS-016-01-0062>
- [Calvo et al., 2018] Calvo, I., Merayo, M., Núñez, M.: ‘AUNTY: A Tool to Automatically Analyze Data Using Fuzzy Automata’; In *2018 3rd International Conference on Computational Intelligence and Applications (ICCIA)* (2018), 102–106. <https://doi.org/10.1109/ICCIA.2018.00026>
- [Calvo et al., 2018] Calvo, I., Merayo, M. G., Núñez, M.: ‘An Improved and Tool-Supported Fuzzy Automata Framework to Analyze Heart Data’; In N. T. Nguyen, D. H. Hoang, T.-P. Hong, H. Pham & B. Trawiński (Eds.), *Intelligent Information and Database Systems*. Cham: Springer International Publishing (2018), 694–704. https://doi.org/10.1007/978-3-319-75417-8_65
- [Calvo et al., 2019] Calvo, I., Merayo, M. G., Núñez, M.: ‘A methodology to analyze heart data using fuzzy automata’; *Journal of Intelligent & Fuzzy Systems*, 37, 6 (2019), 7389–7399. <https://doi.org/10.3233/JIFS-179348>
- [Ceri et al., 2000] Ceri, S., Fraternali, P., Bongio, A.: ‘Web Modeling Language (WebML): a modeling language for designing Web sites’; *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33, 1–6 (2000), 137–157. [https://doi.org/10.1016/S1389-1286\(00\)00040-2](https://doi.org/10.1016/S1389-1286(00)00040-2)
- [Chen et al., 2014] Chen, T., Diciolla, M., Kwiatkowska, M., Mereacre, A.: ‘Quantitative verification of implantable cardiac pacemakers over hybrid heart models’; *Information and Computation*, 236, C (2014), 87–101. <https://doi.org/10.1016/j.ic.2014.01.014>

- [Costa et al., 2019] Costa, T. F., Sobrinho, A., e Silva, L. C., da Silva, L. D., Perkusich, A.: ‘A Coloured Petri Nets Reference Model of Insulin Infusion Pump Control Systems: Assisting the Certification Process’; In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society* (Vol. 1) (2019), 2871–2876. <https://doi.org/10.1109/IECON.2019.8927111>
- [Daly, 2004] Daly, C.: ‘Emfatic. A textual syntax for EMF Ecore (meta-)models’; (2004). Retrieved 1 October 2021, from <https://www.eclipse.org/emfatic/>
- [Dumas et al., 2018] Dumas, M., La Rosa, M., Mendling, J., Reijers, H. A.: ‘Fundamentals of Business Process Management’; Berlin, Heidelberg: Springer Berlin Heidelberg (2018). <https://doi.org/10.1007/978-3-662-56509-4>
- [Fayolle et al., 2016] Fayolle, T., Frappier, M., Gervais, F., Laleau, R.: ‘Modelling a Hemodialysis Machine Using Algebraic State-Transition Diagrams and B-like Methods’; In M. Butler, K.-D. Schewe, A. Mashkoo & M. Biro (Eds.), *Abstract State Machines, Alloy, B, TLA, VDM, and Z*. Cham: Springer International Publishing (2016), 394–408. https://doi.org/10.1007/978-3-319-33600-8_33
- [Gomes and Oliveira, 2009] Gomes, A. O., Oliveira, M. V. M.: ‘Formal Specification of a Cardiac Pacing System’; In A. Cavalcanti & D. R. Dams (Eds.), *FM 2009: Formal Methods*. Berlin, Heidelberg: Springer (2009), 692–707. https://doi.org/10.1007/978-3-642-05089-3_44
- [Gomes and Butterfield, 2016] Gomes, A. O., Butterfield, A.: ‘Modelling the Haemodialysis Machine with Circus’; In *Proceedings of the 5th International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z - Volume 9675*. Berlin, Heidelberg: Springer-Verlag (2016), 409–424. https://doi.org/10.1007/978-3-319-33600-8_34
- [Groenewegen and Visser, 2013] Groenewegen, D. M., Visser, E.: ‘Integration of data validation and user interface concerns in a DSL for web applications’; *Software & Systems Modeling*, 12, 1 (2013), 35–52. <https://doi.org/10.1007/s10270-010-0173-9>
- [Haarmark et al., 2010] Haarmark, C., Graff, C., Andersen, M. P., Hardahl, T., Struijk, J. J., Toft, E., et al.: ‘Reference values of electrocardiogram repolarization variables in a healthy population’; *Journal of Electrocardiology*, 43, 1 (2010), 31–39. <https://doi.org/10.1016/j.jelectrocard.2009.08.001>
- [Herrera-Viedma et al., 2014] Herrera-Viedma, E., Cabrerizo, F. J., Kacprzyk, J., Pedrycz, W.: ‘A review of soft consensus models in a fuzzy environment’; *Information Fusion*, 17 (2014), 4–13. <https://doi.org/10.1016/j.inffus.2013.04.002>
- [Hoang et al., 2016] Hoang, T. S., Snook, C., Ladenberger, L., Butler, M.: ‘Validating the Requirements and Design of a Hemodialysis Machine Using iUML-B, BMotion Studio, and Co-Simulation’; In *Proceedings of the 5th International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z - Volume 9675*. Berlin, Heidelberg: Springer-Verlag (2016), 360–375. https://doi.org/10.1007/978-3-319-33600-8_31
- [Hozo et al., 2005] Hozo, S. P., Djulbegovic, B., Hozo, I.: ‘Estimating the mean and variance from the median, range, and the size of a sample’; *BMC Medical Research Methodology*, 5, 1 (2005), 13. <https://doi.org/10.1186/1471-2288-5-13>
- [Jee et al., 2010] Jee, E., Wang, S., Kim, J. K., Lee, J., Sokolsky, O., Lee, I.: ‘A Safety-Assured Development Approach for Real-Time Software’; In *2010 IEEE 16th International Conference on Embedded and Real-Time Computing Systems and Applications* (2010), 133–142. <https://doi.org/10.1109/RTCSA.2010.42>
- [Jiang et al., 2010] Jiang, Z., Pajic, M., Connolly, A., Dixit, S., Mangharam, R.: ‘Real-Time Heart Model for Implantable Cardiac Device Validation and Verification’; In *2010 22nd Euromicro Conference on Real-Time Systems* (2010), 239–248. <https://doi.org/10.1109/ECRTS.2010.36>

- [Jiang et al., 2010] Jiang, Z., Connolly, A., Mangharam, R.: ‘Using the Virtual Heart Model to validate the mode-switch pacemaker operation’; In 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology (2010), 6690–6693. <https://doi.org/10.1109/IEMBS.2010.5626262>
- [Kauers et al., 2011] Kauers, M., Pillwein, V., Saminger-Platz, S.: ‘Dominance in the family of Sugeno–Weber t-norms’; *Fuzzy Sets and Systems*, 181, 1 (2011), 74–87. <https://doi.org/10.1016/j.fss.2011.04.007>
- [Khan et al., 2018] Khan, S., Hasan, O., Mashkoo, A.: ‘Formal Verification and Safety Assessment of a Hemodialysis Machine’; In A. M. Tjoa, L. Bellatreche, S. Biffl, J. van Leeuwen & J. Wiedermann (Eds.), *SOFSEM 2018: Theory and Practice of Computer Science*. Cham: Springer International Publishing (2018), 241–254. https://doi.org/10.1007/978-3-319-73117-9_17
- [Kolovos et al., 2020] Kolovos, D. S., García-Domínguez, A., Rose, L. M., Paige, R. F.: ‘Epsilon Book’; (2020). Retrieved 1 October 2021, from <https://www.eclipse.org/epsilon/doc/book/>
- [Kühne, 2006] Kühne, T.: ‘Matters of (Meta-) Modeling’; *Software & Systems Modeling*, 5, 4 (2006), 369–385. <https://doi.org/10.1007/s10270-006-0017-9>
- [Kwiatkowska et al., 2014] Kwiatkowska, M., Lea-Banks, H., Mereacre, A., Paoletti, N.: ‘Formal Modelling and Validation of Rate-Adaptive Pacemakers’; In 2014 IEEE International Conference on Healthcare Informatics (2014), 23–32. <https://doi.org/10.1109/ICHI.2014.11>
- [Leemans and Amalio, 2012] Leemans, J., Amalio, N.: ‘Modelling a cardiac pacemaker visually and formally’; Presented at the 2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2012), IEEE Computer Society (2012), 257–258. <https://doi.org/10.1109/VL-HCC.2012.6344542>
- [Loewer and Laddaga, 1985] Loewer, B., Laddaga, R.: ‘Destroying the consensus’; *Synthese*, 62, 1 (1985), 79–95. <https://doi.org/10.1007/BF00485388>
- [Ludewig, 2003] Ludewig, J.: ‘Models in software engineering – an introduction’; *Software and Systems Modeling*, 2, 1 (2003), 5–14. <https://doi.org/10.1007/s10270-003-0020-3>
- [Macià et al., 2016] Macià, H., Valero, V., Díaz, G., Boubeta-Puig, J., Ortiz, G.: ‘Complex Event Processing Modeling by Prioritized Colored Petri Nets’; *IEEE Access*, 4 (2016), 7425–7439. <https://doi.org/10.1109/ACCESS.2016.2621718>
- [Mahulea et al., 2014] Mahulea, C., Mahulea, L., García-Soriano, J.-M., Colom, J.-M.: ‘Petri nets with resources for modeling primary healthcare systems’; In 2014 18th International Conference on System Theory, Control and Computing (ICSTCC) (2014), 639–644. <https://doi.org/10.1109/ICSTCC.2014.6982489>
- [Masci et al., 2013] Masci, P., Ayoub, A., Curzon, P., Lee, I., Sokolsky, O., Thimbleby, H.: ‘Model-Based Development of the Generic PCA Infusion Pump User Interface Prototype in PVS’; In F. Bitsch, J. Guiochet & M. Kaâniche (Eds.), *Computer Safety, Reliability, and Security*. Berlin, Heidelberg: Springer (2013), 228–240. https://doi.org/10.1007/978-3-642-40793-2_21
- [Méry and Singh, 2012] Méry, D., Singh, N. K.: ‘Formalization of Heart Models Based on the Conduction of Electrical Impulses and Cellular Automata’; In Z. Liu & A. Wassylng (Eds.), *Foundations of Health Informatics Engineering and Systems*. Berlin, Heidelberg: Springer (2012), 140–159. https://doi.org/10.1007/978-3-642-32355-3_9
- [Moody and Mark, 2001] Moody, G. B., Mark, R. G.: ‘The impact of the MIT-BIH Arrhythmia Database’; *IEEE Engineering in Medicine and Biology Magazine*, 20, 3 (2001), 45–50. <https://doi.org/10.1109/51.932724>
- [Mordecai, 2019] Mordecai, Y.: ‘Model-based protocol specification’; *Systems Engineering*, 22, 2 (2019), 188–210. <https://doi.org/10.1002/sys.21480>
- [Object Management Group, 2003] Object Management Group: ‘The Common Warehouse Metamodel Specification - version 1.1’; (2003). Retrieved 1 October 2021, from <https://www.omg.org/spec/CWM/About-CWM>

- [Object Management Group, 2017] Object Management Group: ‘The Unified Modeling Language Specification - version 2.5.1’; (2017). Retrieved 1 October 2021, from <https://www.omg.org/spec/UML/About-UML>
- [Object Management Group, 2020] Object Management Group: ‘BPMN Specification - Business Process Model and Notation’; (2020). Retrieved 1 October 2021, from <https://www.bpmn.org/>
- [Object Management Group, 2021] Object Management Group: ‘MetaObject Facility’; (2021). Retrieved 1 October 2021, from <https://www.omg.org/mof/>
- [Pajic et al., 2012] Pajic, M., Jiang, Z., Lee, I., Sokolsky, O., Mangharam, R.: ‘From Verification to Implementation: A Model Translation Tool and a Pacemaker Case Study’; In 2012 IEEE 18th Real Time and Embedded Technology and Applications Symposium (2012), 173–184. <https://doi.org/10.1109/RTAS.2012.25>
- [Palomares et al., 2014] Palomares, I., Martínez, L., Herrera, F.: ‘MENTOR: A graphical monitoring tool of preferences evolution in large-scale group decision making’; Knowledge-Based Systems, 58 (2014), 66–74. <https://doi.org/10.1016/j.knosys.2013.07.003>
- [Perez et al., 2016] Perez, I. J., Cabrerizo, F. J., Alonso, S., Chiclana, F., Herrera-Viedma, E.: ‘Soft Consensus Models in Group Decision Making’; In T. Calvo Sánchez & J. Torrens Sastre (Eds.), Fuzzy Logic and Information Fusion: To commemorate the 70th birthday of Professor Gaspar Mayor. Cham: Springer International Publishing (2016), 135–153. https://doi.org/10.1007/978-3-319-30421-2_10
- [Pérez et al., 2018] Pérez, I. J., Cabrerizo, F. J., Alonso, S., Dong, Y. C., Chiclana, F., Herrera-Viedma, E.: ‘On dynamic consensus processes in group decision making problems’; Information Sciences, 459 (2018), 20–35. <https://doi.org/10.1016/j.ins.2018.05.017>
- [Perić et al., 2014] Perić, B., Gautier, R., Pickard, C. J., Bosiočić, M., Grbić, M. S., Požek, M.: ‘Solid-state NMRNQR and first-principles study of two niobium halide cluster compounds’; Solid State Nuclear Magnetic Resonance, 59–60 (2014), 20–30. <https://doi.org/10.1016/j.ssnmr.2014.02.001>
- [Popovic et al., 2015] Popovic, A., Lukovic, I., Dimitrieski, V., Djukic, V.: ‘A DSL for modeling application-specific functionalities of business applications’; Computer Languages, Systems and Structures, 43, C (2015), 69–95. <https://doi.org/10.1016/j.cl.2015.03.003>
- [Rijnbeek et al., 2014] Rijnbeek, P. R., van Herpen, G., Bots, M. L., Man, S., Verweij, N., Hofman, A., et al.: ‘Normal values of the electrocardiogram for ages 16–90years’; Journal of Electrocardiology, 47, 6 (2014), 914–921. <https://doi.org/10.1016/j.jelectrocard.2014.07.022>
- [Rodrigues da Silva, 2015] Rodrigues da Silva, A.: ‘Model-driven engineering: A survey supported by the unified conceptual model’; Computer Languages, Systems & Structures, 43 (2015), 139–155. <https://doi.org/10.1016/j.cl.2015.06.001>
- [Roelens and Bork, 2020] Roelens, B., Bork, D.: ‘An Evaluation of the Intuitiveness of the PGA Modeling Language Notation’; In S. Nurcan, I. Reinhartz-Berger, P. Soffer & J. Zdravkovic (Eds.), Enterprise, Business-Process and Information Systems Modeling. Cham: Springer International Publishing (2020), 395–410. https://doi.org/10.1007/978-3-030-49418-6_27
- [Seidewitz, 2003] Seidewitz, E.: ‘What models mean’; IEEE Software, 20, 5 (2003), 26–32. <https://doi.org/10.1109/MS.2003.1231147>
- [Selic, 2003] Selic, B.: ‘The pragmatics of model-driven development’; IEEE Software, 20, 5 (2003), 19–25. <https://doi.org/10.1109/MS.2003.1231146>
- [Simic et al., 2021a] Simic, V. and Gokasar, I. and Deveci, M. and Isik, M.: ‘Fermatean Fuzzy Group Decision-Making Based CODAS Approach for Taxation of Public Transit Investments’; IEEE Transactions on Engineering Management (2021), 1–16. <https://doi.org/10.1109/TEM.2021.3109038>

- [Simic et al., 2021b] Simic, V. and Gokasar, I. and Deveci, M. and Karakurt, A.: ‘An integrated CRITIC and MABAC based type-2 neutrosophic model for public transportation pricing system selection’; *Socio-Economic Planning Sciences* (2021), 101157. <https://doi.org/10.1016/j.seps.2021.101157>
- [Singh et al., 2015] Singh, N. K., Wang, H., Lawford, M., Maibaum, T. S. E., Wassyn, A.: ‘Step-wise Formal Modelling and Reasoning of Insulin Infusion Pump Requirements’; In V. G. Duffy (Ed.), *Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management: Ergonomics and Health*. Cham: Springer International Publishing (2015), 387–398. https://doi.org/10.1007/978-3-319-21070-4_39
- [Steinberg et al., 2008] Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: ‘EMF: Eclipse Modeling Framework’; (2nd ed.). Upper Saddle River, NJ: Addison-Wesley (2008).
- [Torkayesh et al., 2021a] Torkayesh A., Deveci M.: ‘A mulTi-noRmalization mUlti-distance aS-sessment (TRUST) approach for locating a battery swapping station for electric scooters’; *Sustainable Cities and Society*, 74 (2021), 103243. <https://doi.org/10.1016/j.scs.2021.103243>
- [Torkayesh et al., 2021b] Torkayesh A., Deveci M., Torkayesh, S., Babae Tirkolae, E.: ‘Analyzing failures in adoption of smart technologies for medical waste management systems: a type-2 neutrosophic-based approach’; *Environmental Science and Pollution Research* (2021). <https://doi.org/10.1007/s11356-021-16228-9>
- [ten Teije et al., 2006] ten Teije, A., Marcos, M., Balsler, M., van Croonenborg, J., Duelli, C., van Harmelen, F., et al.: ‘Improving medical protocols by formal methods’; *Artificial Intelligence in Medicine*, 36, 3 (2006), 193–209. <https://doi.org/10.1016/j.artmed.2005.10.006>
- [Ureña et al., 2015] Ureña, R., Cabrerizo, F. J., Chiclana, F., Herrera-Viedma, E.: ‘GDM-ViewER: A New Tool in R to Visualize the Evolution of Fuzzy Consensus Processes’; In H. Fujita & G. Guizzi (Eds.), *Intelligent Software Methodologies, Tools and Techniques*. Cham: Springer International Publishing (2015), 319–332. https://doi.org/10.1007/978-3-319-22689-7_24
- [Valero et al., 2021] Valero, V., Díaz, G., Boubeta-Puig, J., Macià, H., Brazález, E.: ‘A Compositional Approach for Complex Event Pattern Modeling and Transformation to Colored Petri Nets with Black Sequencing Transitions’; *IEEE Transactions on Software Engineering* (2021), In press. <https://doi.org/10.1109/TSE.2021.3065584>
- [Voelter, 2013] Voelter, M.: ‘DSL Engineering: Designing, Implementing and Using Domain-Specific Languages’; S.I. (2013).
- [Wang et al., 2020] Wang, L., Malik, A., Roop, P. S., Cheng, L. K., Paskaranandavadi, N., Ai, W.: ‘A novel approach for model-based design of gastric pacemakers’; *Computers in Biology and Medicine*, 116 (2020), 103576. <https://doi.org/10.1016/j.combiomed.2019.103576>
- [Wu and Xu, 2012] Wu, Z., Xu, J.: ‘A consistency and consensus based decision support model for group decision making with multiplicative preference relations’; *Decision Support Systems*, 52, 3 (2012), 757–767. <https://doi.org/10.1016/j.dss.2011.11.022>
- [Xu et al., 2017] Xu, Y., Cabrerizo, F. J., Herrera-Viedma, E.: ‘A consensus model for hesitant fuzzy preference relations and its application in water allocation management’; *Applied Soft Computing*, 58 (2017), 265–284. <https://doi.org/10.1016/j.asoc.2017.04.068>

A AUNTY code fragment

Listing 1 shows an excerpt of the file corresponding to the code, required by the AUNTY tool for the analysis of data, generated from the graphical MODELIFY tool.

Listing 1: AUNTY code fragment

```
HAMACHER
gender,bpm,rr,minute,age
s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,
s16,s17,s18,s19,s20,s21,s22,s23,s24,s25,s26,s27,s28,s29,
s30,s31,s32,s33,s34,s35,s36,s37,s38,s39,s40,s41,s42,s43,
s44,s45,s46,s47,s48,s49,s50,s51,s52,s53,s54,s55,s56,s57,
s58,s59,s60,s61,s62,s63,s64,s65,s66,s67,s68,s69,s70,s71,
s72,s73,s74,s75,s76,s77,s78,s79,s80,s81,s82,s83,s84,s85,
s86,s87,s88,s89,s90,s91,s92,s93,s94,s95,s96,s97,s98,s99,
s100,s101,s102,s103,s104,s105,s106,s107,s108,s109,s110,
s111,s112,s113,s114,s115,s116,s117,s118,s119,s120,s121,s122,
s123,s124,s125,s126,s127,s128,s129
s7
s7 s0
?gender gender
HAMACHER
gender=0 [0.0]
ID
s7 s8
?gender gender
HAMACHER
gender=1 [0.0]
ID
s0 s13
?age age
HAMACHER
20<=age<=29 [4.0]
ID
s13 s11
?min minute
True
ID
s13 s17
?end
True
ID
s11 s15
?bpm bpm
HAMACHER
63<=bpm<=83 [10.0]
ID
s15 s15
?rr rr
True
ID
s15 s12
?endrr
True
ID
s12 s13
!ok minute
True
ID
s11 s16
```

```
?bpm bpm
HAMACHER
bpm<=63 [10]
ID
s11 s16
?bpm bpm
HAMACHER
bpm>=83 [10]
ID
s16 s16
?rr rr
HAMACHER
724<=rr<=1020 [148.0]
ID
s16 s12
?endrr
True
ID
s16 s14
?rr rr
HAMACHER
724>=rr [148.0]
ID
s16 s14
?rr rr
HAMACHER
rr>=1020 [148.0]
ID
s14 s14
?rr rr
True
ID
s14 s19
?endrr
True
ID
s19 s13
!Alarm minute
True
ID
```