Medical Fuzzy Control Systems with Fuzzy Arden Syntax

Jeroen S. de Bruin[1,2], Christian Schuh[3], Andrea Rappelsberger[2], and Klaus-Peter Adlassnig[1,2*]

[1] Medexter Healthcare GmbH, Borschkegasse 7/5, 1090 Vienna, Austria

jb@medexter.com

[2] Section for Artificial Intelligence and Decision Support, Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna, Spitalgasse 23, 1090 Vienna, Austria

andrea.rappelsberger@meduniwien.ac.at

klaus-peter.adlassnig@meduniwien.ac.at

[3] IT4Science, IT Systems & Communications, Medical University of Vienna, Spitalgasse 23, 1090 Vienna, Austria

christian.schuh@meduniwien.ac.at

Abstract Arden Syntax is a formal language for representing and processing medical knowledge that is employed by knowledge-based medical systems. In HL7 International's Arden Syntax version 2.9 (Fuzzy Arden Syntax), the syntax was extended by formal constructs based on fuzzy set theory and fuzzy logic, including fuzzy control. These concepts are used to model linguistic and propositional uncertainty – which is inherent to medical knowledge – in a variety of clinical situations. Using these fuzzy methods, we can create medical fuzzy control systems (MFCSs), in which linguistic control rules are used and evaluated in parallel. Their results are aggregated so that gradual transitions between otherwise discrete control states are enabled. In this paper, we discuss the implementation of MFCSs in Fuzzy Arden Syntax. Through code examples from FuzzyArdenKBWean, an MFCS for weaning support in mechanically ventilated patients after cardiac surgery, we illustrate the implementation of fuzzy control.

Keywords: Arden Syntax • Fuzzy Logic • Fuzzy Control • Clinical Decision Support Systems • Weaning from ventilation

Introduction

Arden Syntax is a widely known international standard for computerized knowledge representation and processing that supports the collection, description, and processing of medical knowledge in a machine-executable format. With Arden Syntax, medical rules and procedures can be expressed using algorithmic expressions and conditional statements. The rule sets are known as medical logic modules (MLMs) and usually contain sufficient logic to make at least a single medical decision [1]. Due to the fact that Arden Syntax MLMs can be interconnected and invoke each other, modularized packages for certain clinical decision support tasks can be established [2].

A drawback of earlier versions of Arden Syntax is that the modeling of fuzziness of linguistic clinical terms and uncertainty with respect to clinical conclusions were not intrinsically supported. Such linguistic and propositional uncertainties are, however, inherent to medical knowledge. For example, clinical guidelines are sometimes expressed using linguistic constructs such as "usually" or "often", which are subject to interpretation and lead to inter-rater variability. The same is true of clinical concepts such as "fever", "increased glucose levels", "leukopenia", and many others. As of version 2.9, Arden Syntax supports formal operators for fuzzy sets and fuzzy logic. Hence we will refer to this version as Fuzzy Arden Syntax [3]. Fuzzy sets can be employed to formally model the unsharpness of linguistic clinical concepts in relation to underlying medical data [4]; fuzzy logic can then be used to evaluate logical combinations of declared clinical concepts in order to draw conclusions about more abstract higher-level clinical concepts, and propagate the results through an inference network.

A number of medical applications have been based on fuzzy sets and fuzzy logic [5, 6]. *Fuzzy control* is of special interest in this paper. Medical fuzzy control systems (MFCSs) are based on linguistic control rules. The rules are evaluated in parallel, and their outcome is aggregated such that small transitions between "on" and "off" are possible. Examples of MFCSs include the control of drug dosages for human immunodeficiency virus and acquired immune deficiency syndrome (HIV/AIDS)-infected patients [7], the control of limb prostheses [8], and the regulation of mechanical ventilators in intensive care units [9].

In the present report, we discuss how fuzzy control is intrinsically supported in Fuzzy Arden Syntax. Using examples from FuzzyArdenKBWean, a system for weaning support in mechanically ventilated patients after cardiac surgery [10], we show how fuzzy sets and fuzzy logic control rules can be implemented in Fuzzy Arden Syntax. These MLMs were implemented and executed using the ArdenSuite framework for medical knowledge representation and rule-based inference, which supports Arden Syntax to version 2.10 [11] (including Fuzzy Arden Syntax).

Methods

Arden Syntax

Arden Syntax is a medical knowledge representation and processing standard with properties that make it well suited for the computerized representation of medical knowledge [12]. In Arden Syntax, the program code resembles natural language, thus healthcare professionals can understand the code more easily. Furthermore, it supports data types tailored to the needs of medical documentation, such as data concerning time and duration. Finally, medical knowledge is separated from technical code, which improves code transparency. We will describe Arden Syntax version 2.9 to the extent that the reader will be able to understand the present report and the examples mentioned therein. For a complete description of the syntax we refer to the Arden Syntax version 2.9 specification [3].

In Arden Syntax, knowledge bases are segmented into MLMs. Each MLM is constructed hierarchically. At the top level, an MLM is divided into four categories: maintenance, library, knowledge, and resources. These categories, in turn, are divided into slots. The maintenance category contains metadata on the MLM; it includes self-explanatory slots that describe various aspects of the MLM, such as title, author, or version. The library category is meant to provide contextual and background information about the MLM. Using slots such as purpose, explanation, keywords, citations, and links, the MLM author(s) can describe why the MLM was created, what it does, and refer to external sources. The actual implementation of algorithms and rules takes place in the knowledge category. MLM parameters can be declared in the data slot. Apart from input parameters, MLMs can also acquire data from external sources through curly braces expressions, which allow for dynamic interaction between an MLM and the data-providing host system. The MLM algorithms, rules, or program logic expressions are implemented in the logic slot. Other MLMs can also be invoked. Execution in the logic slot is finished with a concluding statement. If the statement proves to be true, the content of the action slot is executed, such as sending data to an external data source or returning a value. Finally, the conditional resources category allows for the construction of localized messages.

Fuzzy Arden Syntax

In this section, we present a selection of fuzzy extensions implemented in Fuzzy Arden Syntax. This is not a complete overview; for an extended survey of fuzzy methods in Fuzzy Arden Syntax, we refer to previously published work on the subject [13].

An underlying principle of fuzzy methods in Fuzzy Arden Syntax is the extension of the syntax's truth value model. In Fuzzy Arden Syntax, a truth value is defined over a continuous spectrum in a range [0,1] rather than a dichotomous "true/false" model. In this range 0 stands for false, 1 for true, and an intermediate value indicates a degree of truth (or compatibility). Based on this extension, Fuzzy Arden Syntax intrinsically supports fuzziness with data types, built-in propositional fuzzy logic operators, and handling of fuzzy conditions in conditional branches.

With the fuzzy set data type, the unsharpness of boundaries in definitions of linguistic concepts can be conveniently modeled and explicitly calculated. A fuzzy set declaration requires that the boundaries of the fuzzy region be specified. Based on these boundaries, a linear membership function is associated to the variable, which is then able to calculate the truth value of measured data with respect to the clinical linguistic concept under consideration.

Three basic propositional fuzzy logic operations are implemented in Fuzzy Arden Syntax: conjunction, disjunction, and negation. These operations are equipped to handle all truth values in the specified range [0,1]. In Fuzzy Arden Syntax, these operators were implemented by the standard intersection, union, and complement operators *min*, *max*, and *1-x*, respectively [14].

Because of the extended truth value model, it is possible that conditions in conditional branches are neither true nor false. When this happens, the affected conditional branches are executed in parallel; they are also assigned a *degree of applicability* (DoA), which refers to the degree to which it is reasonable to use the value of a variable or set of variables modified or assigned in the respective branch [3]. By default, the DoA equals 1, and is reduced automatically to a weighted average when a program branches on a fuzzy condition; after all, since the condition enabling this program branch was neither true nor false, any values resulting from its execution are relativized accordingly.

In Fuzzy Arden Syntax, the DoA was implemented as follows: When *n* conditional statements are grouped in if-then-elseif statements with fuzzy values as conditions, the execution of the MLM is split into *n* branches, which are executed in parallel. In this process, each branch is provided with its own set of duplicated variables. Furthermore, each branch is assigned a DoA, which is the truth value of its condition divided by the sum of all truth values in the if-then-elseif block.

If, after execution of all conditional branches, the if-then-elseif block is not subsumed using the "*aggregate*" keyword, the different sets of duplicated variables will remain separate and the MLM will conclude with multiple return values, each with a DoA equal to the DoA of its respective program branch. However, if the branches are aggregated, the values of the variables are joined using a weighted average, and only one value for each variable is returned, here with a DoA equal to 1.

As a practical example of using fuzzy constructs in Fuzzy Arden Syntax, consider the MLM code below. Note that we have limited the example to the knowledge category:

```
knowledge:
  type: data_driven;;
  data: (lcnt) := argument;; // Laboratory result
  priority: ;;
  evoke: ;;
  logic:
    // Fuzzy set definitions
    fs_leukopenia := fuzzy set (4000,1), (5000,0);
    fs_leukocytosis := fuzzy set (11000,0), (12000,1);

    // Leukocyte count analysis
    if (lcnt is in fs_leukopenia) or
      (lcnt is in fs_leukocytosis) then
    msg := "Leukocyte count is in pathological range";
    else
      msg := "Leukocyte count is in normal range";
    endif;
    conclude true;;
  action: return msg;;
  urgency: ;;
end:
```

The logic in this MLM is based on infection surveillance criteria defined by the European Centre for Disease Prevention and Control [15]. Surveillance definitions for leukopenia (*4,000 white blood cells (WBC) per mm³ blood* or less) and leukocytosis (*12,000 WBC/mm³ blood* or more) are included in these criteria. However, one might argue that patients with measured values close to these thresholds are also of interest. As such, we created fuzzy sets for both clinical concepts that extend beyond the defined thresholds (Figure 1).

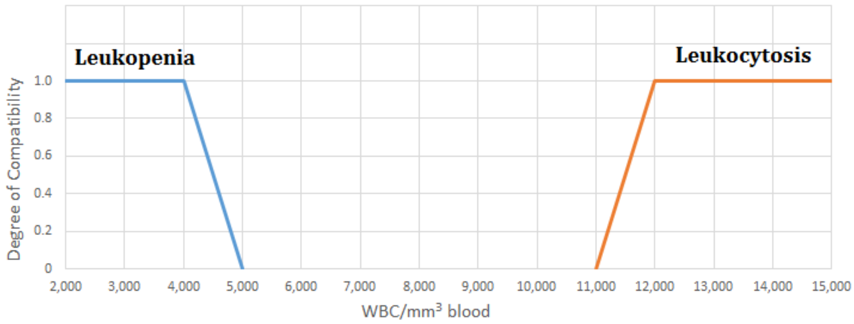

Fig. 1. Graphical depiction of leukopenia and leukocytosis fuzzy sets. Note: WBC, white blood cells.

Truth values are determined during leukocyte count analysis with both fuzzy sets. The truth values are then combined using a logical fuzzy disjunction. In case the outcome is neither true nor false, both conditional branches are executed; as the branches are not aggregated, this would cause the MLM to return two copies of *msg*, each with its own DoA. For example, if the laboratory result were to be *4,400 WBC/mm*^{3}, the outcome of the conditional statement would be *0.6*, due to the evaluation with *fs_leukopenia*. Consequentially, the DoA of that conditional branch would be *0.6*, and for the *else* branch it is automatically *0.4*. As no *aggregate* keyword was provided at the *endif* branch closure, two copies of the *msg* variable are returned: one with a DoA of *0.6* that specifies that "*leukocyte count is in pathological range*", and another with a DoA of *0.4* that states that "*leukocyte count is in normal range*".

Fuzzy Control and FuzzyArdenKBWean

In fuzzy control, the control strategy is written in "if-then-else" statements similar to natural language rather than using abstract mathematical equations. In these statements, the unsharpness of linguistic terms is represented by fuzzy sets. As a result, transition between control states in fuzzy control systems is more gradual than in traditional control systems.

In general, a fuzzy controller performs the following actions: First, (discrete) system inputs are acquired from the device to be controlled and possibly from additional data sources. The inputs are then interpreted with fuzzy sets (fuzzification) to produce truth values for linguistic concepts. Using the resulting truth values as conditions, all linguistic control rules in the knowledge base are evaluated in parallel, yielding a set of values for each output parameter. In the last processing step, values in each set are aggregated and made discrete again (defuzzification). These discrete results are then either interpreted by the user to manually adjust the system (open-loop system), or fed back into the system itself (closed-loop system).

FuzzyArdenKBWean is an open-loop MFCS that works as described above. This system was developed to improve weaning support in mechanically ventilated patients after cardiac surgery in intensive care units. More specifically, the system optimizes the weaning process (i.e., the transition from full to no ventilation support) by trying to achieve optimal values for arterial oxygen partial pressure (P_aO_2), arterial carbon dioxide partial pressure (P_aCO_2), and the fraction of inspired oxygen (F_iO_2). It is an open-loop, knowledge-based control system that proposes changes to peak inspiratory pressure (PIP), positive end expiratory pressure (PEEP), and F_iO_2, which patient caregivers then implement or, due to reasons unknown to the MFCS, deviate from these proposals.

On average, measured in our tests, FuzzyArdenKBWean reacted 131 minutes earlier than the attending physicians, with a standard error of mean (SEM) of 47 minutes. The mean delay in case of hyperventilation was 127 minutes, (SEM 34); the corresponding value for hypoventilation was 50 minutes (SEM 21).

ArdenSuite

In order to obtain examples for the present report, MLMs in FuzzyArdenKBWean were created with the ArdenSuite software [11, 16]. ArdenSuite is a framework for medical knowledge representation, rule-based inference, and extended integration in health IT landscapes. It comprises an integrated development and test environment (ArdenSuite IDE) and the ArdenSuite Server, including software to interconnect with data sources (Figure 2).

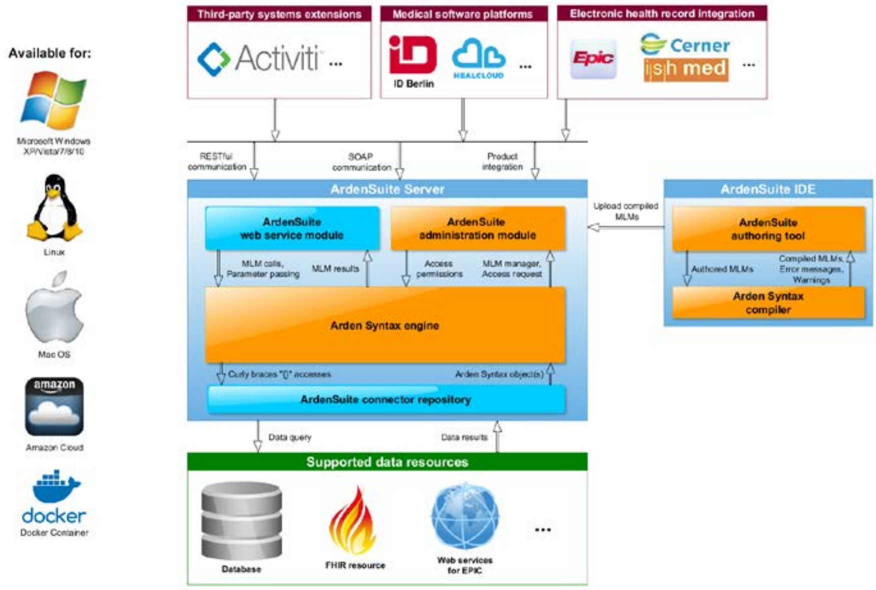

Fig. 2. The ArdenSuite framework for medical knowledge representation, rule-based inference, and health IT integration.

The ArdenSuite IDE includes an authoring component, which allows users to write and compile MLMs. Given the appropriate data, the IDE also allows users to immediately test the written MLMs. After MLMs have been compiled, they are uploaded to the ArdenSuite server. The central element of the ArdenSuite server is the Arden Syntax engine, which executes the compiled MLMs. On top of the engine, an administration module is provided. Functionality reaches from being a repository for compiled Arden Syntax projects to allowing for the management of those projects (such as MLM version management, activation or deactivation of

MLMs in an application, or implementing temporal restrictions). Furthermore, the server hosts a web service component that enables service-oriented access to the server by arbitrary clients.

To promote interoperability between the ArdenSuite server and host systems, such as electronic health records (EHRs), the system is provided with several forms of server and data access as well as multiple communication standards. For data exchange and MLM execution, the ArdenSuite server supports various web services. MLM and event calls are realized by Simple Object Access Protocol (SOAP) or Representational State Transfer (REST); the data required for MLM processing can also be provided in this call. Alternatively, data can also be acquired from external databases, Fast Healthcare Interoperability Resources (FHIR) resources, and through web services for EPIC using the ArdenSuite connector repository.

Results

The first step in implementing FuzzyArdenKBWean was the construction of fuzzy sets for classifying the inputs in linguistic terms. Based on medical experience, as well as statistical data from prospective randomized trials and archived data, fuzzy sets were created for linguistic classifications (Very low, Low, Normal, High, and Very high, respectively) for both inputs (Figure 3).

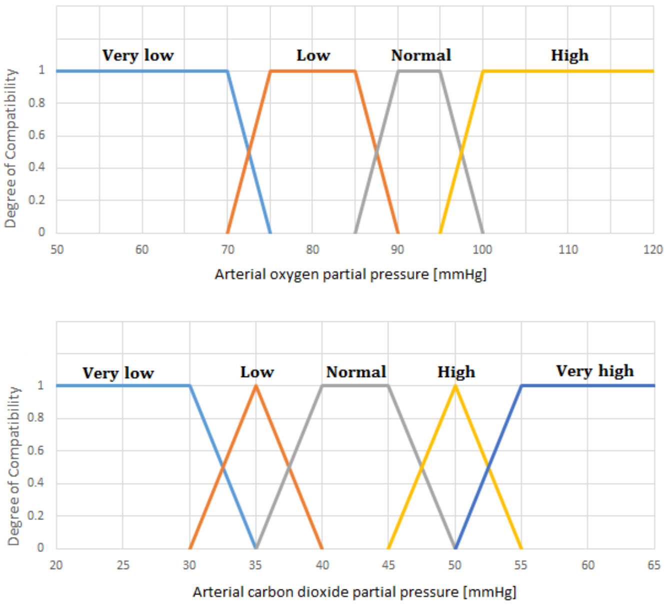

Fig. 3. Graphical depiction of fuzzy sets defined in FuzzyArdenKBWean.

The fuzzy sets in Figure 3 were used as conditions of fuzzy control rules in the knowledge base. The following rules were selected for this paper:

R1: IF O_2 IS NORMAL AND CO_2 IS VERY HIGH THEN PIP = +5;
R2: IF O_2 IS LOW AND CO_2 IS VERY HIGH THEN PIP = +5;
R3: IF O_2 IS LOW AND CO_2 IS HIGH THEN PIP = +0;
R4: IF O_2 IS NORMAL AND CO_2 IS HIGH THEN PIP = +0;

Note that despite rules R3 and R4 seem to have no impact, their presence is vital in the control mechanism, as they influence the final PIP increase, depending on the truth value of their condition.

Finally, for defuzzification we used the aggregate command. The resulting (partial) MLM code is shown below. Due to space constraints, we only defined a subset of the fuzzy sets used in the rule selection mentioned earlier, namely those that were used in conditions in this MLM:

```
knowledge:
  type: data-driven;;
  data: (O2, CO2) := argument;; // PaO2 and PaCO2
  priority: ;;
  evoke: ;;
  logic:
    // Fuzzy set definitions
    O2_low := FUZZY SET (70,0), (75,1), (85,1), (90,0);
    O2_normal := FUZZY SET (85,0), (90,1), (95,1), (100,0);
    CO2_high := FUZZY SET (45,0), (50,1), (55,0);
    CO2_very_high := FUZZY SET (50,0), (55,1);

    // Rule analysis
    if (O2 is in O2_normal) and (CO2 is in CO2_very_high) then
      PIP_inc := 5; // R1
    elseif (O2 is in O2_low) and (CO2 is in CO2_very_high) then
      PIP_inc := 5; // R2
    elseif (O2 is in O2_low) and (CO2 is in CO2_high) then
      PIP_inc := 0; // R3
    elseif (O2 is in O2_normal) and (CO2 is in CO2_high) then
      PIP_inc := 0; // R4
    endif aggregate;
    conclude true;
  ;;
  action:
    return PIP_inc;;
  urgency: ;;
end:
```

To clarify the workings of this MLM, let us consider an example. Assume that PaO2 equals *89* and PaCO2 equals *52*. This yields the following truth values for the defined fuzzy sets: (O2_normal, *0.8*), (O2_low, *0.2*), (CO2_high, *0.6*), and (CO2_very_high, *0.4*). Using the standard intersection operator, the truth values for the rule conditions evaluate to: (R1, *min(0.8, 0.4)=0.4*), (R2, *min(0.2, 0.4)=0.2*), (R3, *min(0.2, 0.6)=0.2*), and (R4, *min(0.8, 0.6)=0.6).* Given that the total sum of truth values for these conditions equals (*0.4+0.2+0.2+0.6=1.4*), the weighted average of individual condition truth values, thus the DoA for the conditional branches, is: (R1, *(0.4 / 1.4) = 0.285*), (R2, *(0.2 / 1.4) = 0.143*), (R3, *(0.2 / 1.4) = 0.143*), (R4, *(0.6 / 1.4) = 0.429*).

Finally, the *aggregate* keyword causes a defuzzification of all the program branches and different *PIP_inc* copies, resulting in a single *PIP_inc* value:

$$PIP_{inc} = (0.285 * 5) + (0.143 * 5) + (0.143 * 0) + (0.429 * 0) = 2.14$$

Thus, the program will return the suggestion that the ventilator's PIP should be increased by 2.14 percentage points.

Discussion

In the present report, we showed how MFCSs can be implemented using fuzzy methods supported by Arden Syntax version 2.9, an international HL7 standard for computerized knowledge representation and processing that incorporates fuzzy methods. This is important because fuzzy control is used increasingly often in medical devices and systems and has yielded encouraging results [6]. As logical rules are implemented in natural language, clinical experts together with clinical knowledge engineers can implement their expertise quite easily without having to learn complex syntaxes of current programming languages. Uncertainty and incompleteness of knowledge can be modeled by introducing fuzzy sets for linguistic concepts that are part of these rules. Through the ArdenSuite server, with its standardized communication as well as information exchange capabilities, MFCSs can be more easily integrated into small and large healthcare institutions or single medical applications.

In our experience, the implementation of FuzzyArdenKBWean in Fuzzy Arden Syntax yields the MLMs to be clearer and easier to understand in comparison to implementations done with functional programming languages (Delphi or Java). Although clinicians indicated that they were not able to rapidly produce MLMs by themselves, they did find it straightforward and easy to validate written MLMs and identify logical flaws. As such, the cooperation between knowledge engineer and clinician becomes more productive, resulting in high quality medical software.

The limitations of the present report are worthy of note. The study is limited to a single application, FuzzyArdenKBWean. Other forms of fuzzy control in medicine, e.g., those mentioned in [5, 6], need to be studied to see whether those can be implemented in Fuzzy Arden Syntax as well. For example, for defuzzification we used a weighted average based on truth values and the DoA. Other ways, such as centroid methods or mean–max methods need to be studied too. Furthermore, we have not yet fully tested how the program behaves in real time. Such performance is crucial, especially in intensive care units.

We performed the first steps in implementing fuzzy control with Fuzzy Arden Syntax. In the future, we plan to study and address aforementioned limitations, and also apply Fuzzy Arden Syntax for other medical areas and tasks, such as fuzzy automata for real-time monitoring purpose.

References

1. Hripcsak G (1994) Writing Arden Syntax Medical Logic Modules. Comput Biol Med 24(5):331–363.
2. Adlassnig K-P, Rappelsberger A (2008) Medical Knowledge Packages and Their Integration into Health-Care Information Systems and the World Wide Web. Stud Health Technol Inform 136:121–126.
3. Health Level Seven International (2013) HL7 Arden V2.9-2013: The Arden Syntax for Medical Logic Systems Version 2.9. Available at: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=290. Accessed 27 March 2017.
4. Adlassnig K-P (1988) Uniform Representation of Vagueness and Imprecision in Patient’s Medical Findings Using Fuzzy Sets. In: Trappl R (ed) Cybernetics and Systems’88. Kluwer Academic Publishers, Dordrecht, pp. 685–692.
5. Abbod MF, von Keyserlingk DG, Linkens DA, Mahfouf M (2001) Survey of Utilisation of Fuzzy Technology in Medicine and Healthcare. Fuzzy Sets Syst 120(2):331–349.
6. Mahfouf M, Abbod MF, Linkens DA (2001) A Survey of Fuzzy Logic Monitoring and Control Utilisation in Medicine. Artif Intell Med 21(1-3):27–42.
7. Assawinchaichote W (2015) Control of HIV/AIDS Infection System with Drug Dosages Design via Robust H_∞ Fuzzy Controller. Biomed Mater Eng 26(Suppl 1):S1945–S1951. doi: 10.3233/BME-151497
8. Yang P, Yue H, Chen L, Geng Y (2012) Intelligent Lower Limb Prosthesis Following Healthy Leg Gait Based on Fuzzy Control. In: 2012 24th Chinese Control and Decision Conference (CCDC). IEEE, New York, pp. 3729–3731. doi: 10.1109/CCDC.2012.6244597
9. Schuh C (2007) Managing Uncertainty with Fuzzy-Automata and Control in an Intensive Care Environment. In: Castillo O, Melin P, Montiel Ross O, Sepúlveda Cruz R, Pedrycz W, Kacprzyk J (eds) Theoretical Advances and Applications of Fuzzy Logic and Soft Computing. ASC, vol 42. Springer, Berlin, pp. 263–271.
10. Schuh C, Hiesmayr M, Kaipel M, Adlassnig K-P (2004) Towards an Intuitive Expert System for Weaning from Artificial Ventilation. In: Dick S, Kurgan L, Musilek P, Pedrycz W, Reformat M (eds) NAFIPS 2004 – Annual Meeting of the North American Fuzzy Information Processing Society: Fuzzy Sets in the Heart of the Canadian Rockies, Vol. 2. IEEE, Piscataway, pp. 1008–1012. doi:10.1109/NAFIPS.2004.1337445
11. Medexter Healthcare (2015) ArdenSuite – Medical Knowledge Representation and Rule-Based Inference Software with Arden Syntax. Available at:

http://www.medexter.com/component/jdownloads/send/3-public-articles/6-ardensuite-for-emrs. Accessed 27 March 2017.
12. Samwald M, Fehre K, de Bruin J, Adlassnig K-P (2012) The Arden Syntax Standard for Clinical Decision Support: Experiences and Directions. J Biomed Inform 45(4):711–718. doi:10.1016/j.jbi.2012.02.001
13. Vetterlein T, Mandl H, Adlassnig K-P (2010) Processing Gradual Information with Fuzzy Arden Syntax. Stud Health Technol Inform 160:831–835.
14. Michael D (1959) A Propositional Calculus with Denumerable Matrix. The Journal of Symbolic Logic 24(2):97–106.
15. European Centre for Disease Prevention and Control (ECDC) (2015) European Surveillance of Healthcare-Associated Infections in Intensive Care Units, HAI-Net ICU Protocol, Protocol Version 1.02. Available at: http://ecdc.europa.eu/en/publications/Publications/healthcare-associated-infections-HAI-ICU-protocol.pdf. Accessed 27 March 2017.
16. Adlassnig K-P, Fehre K (2014) Service-Oriented Fuzzy-Arden-Syntax-Based Clinical Decision Support. Indian J Med Inform 8(2):75–79.