Artificial Intelligence in Medicine 92 (2018) 71–81

Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Executable medical guidelines with Arden Syntax—Applications in dermatology and obstetrics

Alexander Seitinger[a], Andrea Rappelsberger[b], Harald Leitich[c], Michael Binder[d], Klaus-Peter Adlassnig[a,b,*]

[a] Medexter Healthcare GmbH, Borschkegasse 7/5, A-1090 Vienna, Austria
[b] Section for Artificial Intelligence and Decision Support, Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna, Spitalgasse 23, A-1090 Vienna, Austria
[c] Department of Obstetrics and Gynecology, Medical University of Vienna, Währinger Gürtel 18-20, A-1090 Vienna, Austria
[d] "Strategic Planning and Control", Health Care Management, The Vienna Hospital Association, Thomas-Klestil-Platz 7/1, A-1030 Vienna, Austria

ARTICLE INFO

Article history:
Received 17 May 2015
Accepted 10 August 2016

Keywords:
Clinical decision support
Arden Syntax
Fuzzy set theory and logic
Medical practice guidelines
Standard operating procedures
Dermatology
Lyme borreliosis
Obstetrics

ABSTRACT

Introduction: Clinical decision support systems (CDSSs) are being developed to assist physicians in processing extensive data and new knowledge based on recent scientific advances. Structured medical knowledge in the form of clinical alerts or reminder rules, decision trees or tables, clinical protocols or practice guidelines, score algorithms, and others, constitute the core of CDSSs. Several medical knowledge representation and guideline languages have been developed for the formal computerized definition of such knowledge. One of these languages is Arden Syntax for Medical Logic Systems, an International Health Level Seven (HL7) standard whose development started in 1989. Its latest version is 2.10, which was presented in 2014. In the present report we discuss Arden Syntax as a modern medical knowledge representation and processing language, and show that this language is not only well suited to define clinical alerts, reminders, and recommendations, but can also be used to implement and process computerized medical practice guidelines.
Methods: This section describes how contemporary software such as Java, server software, web-services, XML, is used to implement CDSSs based on Arden Syntax. Special emphasis is given to clinical decision support (CDS) that employs practice guidelines as its clinical knowledge base.
Results: Two guideline-based applications using Arden Syntax for medical knowledge representation and processing were developed. The first is a software platform for implementing practice guidelines from dermatology. This application employs fuzzy set theory and logic to represent linguistic and propositional uncertainty in medical data, knowledge, and conclusions. The second application implements a reminder system based on clinically published standard operating procedures in obstetrics to prevent deviations from state-of-the-art care. A to-do list with necessary actions specifically tailored to the gestational week/labor/delivery is generated.
Discussion: Today, with the latest versions of Arden Syntax and the application of contemporary software development methods, Arden Syntax has become a powerful and versatile medical knowledge representation and processing language, well suited to implement a large range of CDSSs, including clinical-practice-guideline-based CDSSs. Moreover, such CDS is provided and can be shared as a service by different medical institutions, redefining the sharing of medical knowledge. Arden Syntax is also highly flexible and provides developers the freedom to use up-to-date software design and programming patterns for external patient data access.

1. Introduction

Medical knowledge is advancing and expanding very rapidly. The worldwide medical community is constantly sharing its new scientific findings and conclusions. Clinicians responsible for patient care have overloaded time schedules and find it difficult

* Corresponding author: Univ.-Prof. DI Dr. Klaus-Peter Adlassnig, FACMI, Section for Artificial Intelligence and Decision Support, Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna, Spitalgasse 23, A-1090 Vienna, Austria.
E-mail address: klaus-peter.adlassnig@meduniwien.ac.at (K.-P. Adlassnig).

https://doi.org/10.1016/j.artmed.2016.08.003
0933-3657/

to keep pace with the most recent scientific data. Evidence-based medical practice guidelines have been developed to support clinicians and assist them in patient-specific clinical decision-making. These guidelines evaluate and summarize the existing evidence for important clinical areas, and provide the latest information on prevention, risk, diagnosis, therapy, prognosis, and cost-effectiveness. Even in their simplest form—as plain text-based documents—guidelines usually provide this information in some procedural form such as if-then statements, tables, or decision trees, which are then read and used in clinical routine. However, text-based guidelines are time consuming to read, may not be available when needed, and often difficult to update. Moreover, a variety of guidelines established by different organizations in different countries may exist for one and the same medical topic, recommending a variety of diagnostic and treatment procedures. These guidelines may be incomplete or overlapping, thus causing further ambiguity.

Software-based executable medical practice guidelines are used to provide relevant medical knowledge in fully computerized form, guiding clinicians through the decision-making process and dispensing with the need to read the respective text-based documents [1,2]. The time taken to disseminate new medical knowledge in clinical practice is reduced, and the new medical knowledge is available when needed. When given access to electronic medical records (EMRs), guideline-based clinical decision support systems (CDSSs) can automatically retrieve patient data, thus rendering additional manual input unnecessary. While simple computerized guidelines such as specialized input forms, dialog systems, or pop-up wizards have the disadvantage that any changes in the knowledge base require changes in the source code representation as well, CDSSs try to solve this problem by separating pure medical knowledge from the system's source code.

An appropriate medical knowledge representation or guideline representation language is needed to model and formalize medical knowledge. Several guideline representation languages are available for this purpose [3,4]. The majority of them were developed independently of each other and are based on different approaches with distinct features.

Arden Syntax [5–7] is a Health Level Seven (HL7) International standard [8] for medical knowledge representation and processing. It divides knowledge into modules (so-called medical logic modules or MLMs), where each module represents the necessary medical knowledge to make at least one clinical decision. Arden Syntax was developed to implement simple alerts, reminders, and recommendations. Later then, Arden Syntax was also focused on sharing so-called knowledge packages. The aim was that only a few changes, or none at all, are needed to share such packages between institutions [7].

PROforma was designed to facilitate the creation of safe and robust guidelines. It is an executable process modeling language and is capable of capturing the logical and procedural content of a protocol in the form of a constraint-satisfaction graph, where nodes represent tasks that can be interpreted or enacted by a computer. Guidelines are modeled as a set of clinical tasks (actions, enquiries, and decisions), which are hierarchically subdivided into plans [9,10].

Asgaard/Asbru [11] focused on the development of a modeling language for the representation of skeletal plans with respect to time, specifying processes and activities. It emphasizes guideline intentions rather than action prescriptions alone. Several options are available for modeling the clinical workflow in a feasible and practical manner [12].

The guideline interchange format GLIF was designed to make it easier to share guidelines between different institutions and software systems [10,13,14]. GLIF tries to build on the most useful features of other guideline models, and incorporate standards used in healthcare. Its expression language was originally based on Arden Syntax, and its default medical data model is based on the HL7 Reference Information Model (RIM) [15].

When comparing these languages in the published literature, Arden Syntax is considered too limited or unfit for the implementation of medical guideline-based CDSSs [10,11,13,16–19]. Some of these articles or book sections refer to sources in which outdated versions of Arden Syntax were used; newer Arden Syntax features are not taken into account. The most recent Arden Syntax version is version 2.10, which was fully implemented by [20]. The purpose of the present report is to reintroduce Arden Syntax and the available tools as a modern representation and processing language for medical knowledge, well suited to implement guideline-based CDSSs as well.

2. Methods

2.1. Implementation analysis

A first step in dealing with guideline-based CDSSs is to convert the existing medical practice guidelines into computer-interpretable formats. Since these guidelines are largely available as unstructured, non-formalized texts (see Fig. 1 from [21]), an automated conversion process is usually not possible. A structured and more formal knowledge representation can be achieved by creating flow charts manually. This knowledge format promotes the comprehensibility of the represented decision-making process and clearly shows whether all of its parts are covered in sufficient detail. It also helps to identify ambiguities and missing information. When several medical guidelines exist for one and the same topic—such as Lyme borreliosis [22–25]—merging the medical knowledge of such guidelines constitutes an additional layer of complexity. Again, for such cases, flowcharts are an appropriate tool to analyze points of concurrence, overlap, or contradiction between the guidelines (see Fig. 2). In addition to facilitating the identification of potential ambiguities and contradictions in medical guidelines, the use of detailed flowcharts as intermediate representation was also found to simplify the transformation of medical guidelines into computer-interpretable Arden Syntax (see Fig. 3).

2.2. Execution environment

An Arden Syntax server, together with its Arden Syntax engine, permits the implementation of CDSSs using a client-server architecture [26,27]. The clients rely on one or more servers for resource and processing power. Following this model, the Arden Syntax server is used to store and process computerized medical practice guidelines in the form of several usually interwoven MLMs, and provides access for administrative and maintenance tasks as well. A web-based client is used to communicate with the Arden Syntax server; here various service interfaces are offered. At present, representational state transfer (REST) and simple object access protocol (SOAP) web-services are supported. REST and SOAP are accessible with tools like client uniform resource locator (cURL) [28] and the PHP scripting language [29], but can also be implemented with other programming or scripting languages that are able to communicate via those services, such as HTML5, Java, or C++.

The present Arden Syntax server strongly promotes service-oriented architectures in medicine. CDS is offered to clinical institutions and can be shared as a service.

2.3. Implementing MLMs

MLMs in Arden Syntax contain the necessary structured knowledge of CDSSs. Arden Syntax has been kept as simple as possible [30], yet powerful enough to implement complex constructs [7]. It

Diagnostic evaluation

Clinical features of LNB

Neurological symptoms usually occur 1–12 (mostly 4–6) weeks after the tick bite, and mainly from July to December. Only 40–50% of the patients can recall a tick bite, and 20–30% report a local skin infection (erythema migrans) (stage I) [11,12]. More than 95% can be classified as early LNB (stage II), defined as signs and symptoms lasting for <6 months. Less than 5% have late LNB (stage III) with duration between 6 months and several years [12]. The natural course of early LNB is often self-limiting [13], whereas late LNB has a chronic course that probably reflects persistent survival of bacteria in nervous tissue.

Table 1 Classification of Lyme neuroborreliosis (LNB)

Early LNB
Neurological symptoms for <6 months
With manifestations confined to PNS (cranial nerves, spinal roots or peripheral nerves) (Bannwarth syndrome)
With CNS manifestations
Late LNB
Neurological symptoms for more than 6 months
With PNS manifestations
With CNS manifestations

PNS, peripheral nervous system.

clonus–myoclonus, ocular flutter, apraxia, hemiparesis or Parkinson-like symptoms have been associated with *Bb* infection [14]. Poliomyelitis-like syndromes [15], and acute stroke-like symptoms caused by cerebral vasculitis [16] are rare and have been documented only in single case reports.

Late LNB (Table 1)
Late neurological manifestations are also entitled 'chronic neuroborreliosis'.

PNS manifestations. It may consist of mononeuropathy, radiculopathy and polyneuropathy [17,18]. In Europe, late polyneuropathy has only been observed in combination with acrodermatitis chronica atrophicans (ACA) [19] – the typical dermatological manifestation during late stage III of borrelial infection – whilst isolated cases of distal symmetric polyneuropathy as a result of a borrelial infection have been reported in American patients [20]. It is of note that a causative relationship between polyneuropathy and borrelial infection cannot be based on the sole detection of *Bb*-specific antibodies in patients with polyneuropathy as those antibodies can also be found in 5–25% of healthy persons [21].

CNS manifestations. It includes (i) cerebral vasculitis and (ii) chronic progressive Lyme encephalitis or encephalomyelitis with tetraspastic syndrome, spastic–ataxic gait disorder and disturbed micturition [18].

Fig. 1. An example of an unstructured medical practice guideline taken from an article in the European Journal of Neurology, Mygland et al., 2010, pp. 8–16). This excerpt reports on the clinical symptoms of Lyme borreliosis, such as erythema migrans, mononeuropathy, and radiculopathy.

allows a similar code style as other general purpose programming languages, and also supports a more human readable writing style.

One of the most difficult tasks of representation languages is to devise a method of obtaining external data without being too tightly coupled with a specific data source system. In Arden Syntax, external data is either directly provided during the evoke process of the MLMs, or read using so-called curly brace expressions. The content of those curly brace expressions is—intentionally—not specified by the Arden Syntax standard. In the published literature, this is referred to as the "curly braces problem" and is seen as a weakness of Arden Syntax [6,17,31–34]. However, this approach provides the opportunity to access external data by implementing Java interfaces or so-called data interface connectors. These connectors are programmed such that they retrieve the results of the executed curly brace expressions and convert them into data objects, which can then be processed by the Arden Syntax server.

One approach to avoid the "curly braces problem" is to provide all necessary data at the time of calling an MLM. The advantage is that MLMs need not be modified when used by other institutions. The responsibility of handling data is shifted to the external client systems. Its disadvantage is that the external systems have to precisely prepare the data needed by each MLM.

Another approach is to avoid using actual database queries in curly brace expressions when implementing MLMs. The content of curly brace expressions should only define the type of required data (by keywords or a JavaScript Object Notation (JSON) [35] object, for instance). The execution engine sends the curly braces expression request to the appropriate connector, which can then map the expressions to actual database queries or use other ways of retrieving the requested data (such as simply reading a text file or calling an algorithm). By doing this, an institution with a different database or way of accessing the required data needs to implement just one "general" data interface connector.

A new feature of Arden Syntax version 2.9 is the use of fuzzy sets, fuzzy logic, and other fuzzy language constructs. Fuzzy sets and logic, and additional new concepts such as fuzzy data types and fuzzy operators extend the existing methods by processing linguistic and propositional uncertainty [26,36]. Instead of merely processing sharply bounded 'yes' or 'no' items, Arden Syntax is now able to compute uncertainty, allowing clinicians to select degrees of certainty as to which symptoms apply, diagnoses are indicated, or therapies demanded. Instead of answering that the patient has fever, the clinician may specify the compatibility of the term *fever* with the measured body temperature and thus define natural gradations from normal to pathological more accurately (cf. Fig. 4).

3. Results

Based on two implemented prototypes, we show how Arden Syntax can be used for guideline-based CDS.

First, a guideline-based recommendation system for the diagnosis and treatment of Lyme borreliosis was developed [37,38]. In this prototype we focused on the communication process between the knowledge base MLMs on the server and the client in order to facilitate reusability of the client software for other medical guidelines.

The second prototype is a guideline-based reminder system in obstetrics. It employs four standard operating procedures (SOPs) published at the Department of Obstetrics and Gynecology, Vienna General Hospital [39,40].

3.1. Diagnosis and treatment recommendations for Lyme borreliosis

In this system, patient data are entered manually using a web-based user interface. After processing the data by applying the defined guideline-based MLM knowledge, diagnosis and treatment recommendations are generated. (Data could have been read from data sources such as an EMR system by using curly brace expressions, but we did not implement this possibility.) The

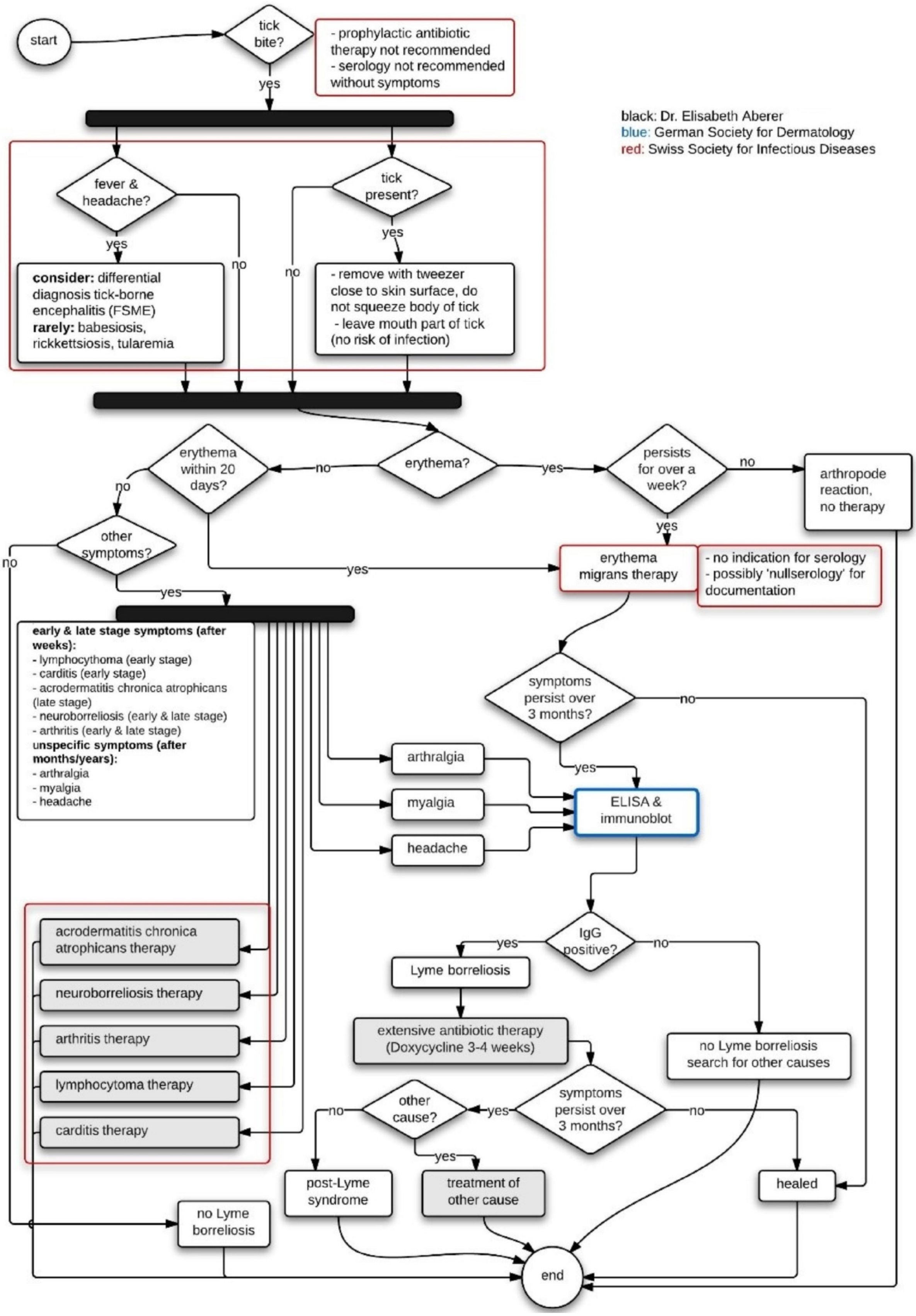

Fig. 2. Excerpt of the finally established Lyme borreliosis guideline, created by analyzing and combining the three available clinical practice guidelines from Austria, Germany, and Switzerland.

```
        question.q_type :=  "polar_fuzzy";
        question.q_label := "Erythema migrans";
        question.q_info := "Erythema with darker margin expanding at the site of the tick bite. Alternate definition: Usually roun

        resultobj.result := resultobj.result,  localized 'erythema';
        resultobj.next_questions :=  resultobj.next_questions, question;
    else
        if erythema_migrans then
            if erythema_migrans_persisted eq -1 then
                question := new question_obj;
                question.q_type :=  "polar_bool";
                question.q_label := "Erytheam migrans persisted";
                question.q_info := "Erythema with darker margin expanding at the site of the tick bite. Alternate definition: Usua

                resultobj.result := resultobj.result,  localized 'erythema_persisted';
                resultobj.next_questions :=  resultobj.next_questions, question;
            elseif erythema_migrans_persisted then
                // erythema for over a week - start therapy
                // -----------------------------------------
                // -----------------------------------------
                if neurologic_symptoms eq -1 then // if neurologic_symptoms = -1 so is carditis
                    question1 := new question_obj;
                    question1.q_type :=  "polar_fuzzy";
                    question1.q_label := "Neurologic symptoms";
                    question1.q_info := "Describe possible symptoms";

                    question2 := new question_obj;
                    question2.q_type :=  "polar_fuzzy";
                    question2.q_label := "Carditis";
                    question2.q_info := "Carditis is the inflammation of the heart or its surroundings.";

                    resultobj.result := resultobj.result,  localized 'check_neuro_symptoms', localized 'check_carditis';
                    resultobj.next_questions :=  resultobj.next_questions, question1, question2;
                elseif neurologic_symptoms then
                    resultobj := call MLM_neuroborreliosis_treatment with parameterlist;
                elseif carditis then
                    resultobj := call MLM_neuroborreliosis_treatment with parameterlist;
                else
                    resultobj := call MLM_erythema_migrans_therapy with parameterlist;
                endif;
            else
                // erythema vanished
                resultobj.result := resultobj.result, localized 'healthy';
            endif;
        else // no erythema, watchful wait
```

Fig. 3. Excerpt of a written Arden Syntax MLM. This MLM portion defines the results to be generated when certain conditions are true.

communication process between the server and its clients may be divided into four steps (see Fig. 5).

First, the client retrieves the available patient data from an internal database selected during a previous session. Second, a connection to the Arden Syntax server is established, requesting the execution of a specific MLM and sending the retrieved patient data as payload. This payload is structured as an Arden Syntax list containing key–value pair objects. While the value may be of any type, each key has to be a string. The executed MLM uses the received patient data as input to traverse the implemented decision tree. Whenever the decision tree checks for conditions, the parameter list is used to look up the required data. If the required data is available, the process continues and recommendations are generated. If the required data is not available, a question object asking for further information is generated, while the current execution of the MLM stops. In a third step, the Arden Syntax server sends the generated answer, containing at least one question or answer object, back to the client as a transfer object (see Fig. 6) with a predefined structure. Since no direct access to the medical data of EMRs was incorporated into this prototype, all required data have to be obtained through manual user input. When integrating the system into or connecting it with an EMR, data can be directly read from the EMR's data base, thus requiring minimal or no manual input.

In a fourth step, the client generates the user interface utilizing this transfer object. In case of user interaction, all additional patient data provided during this interaction are stored in the database before the entire process is repeated. This permits the user to stop the process at any time and resume whenever new data is available. Thus, the system is able to support guidelines spanning longer periods of time, such as multiple patient consultations.

The client only has to be aware of the structure of the transfer object used by the executed MLM in order to be able to represent it. This object is kept simple and very general, thus permitting the implementation of different medical guidelines. Both, the Arden Syntax server answer and the client request employ JSON as the interchange format. The result list contains recommendations generated by the MLM during its execution, and is displayed by the client. Question objects constitute data needed to further traverse the implemented decision tree.

A question object contains a question type, a label, and explanatory details. With this information the client determines the layout of the actual graphical user interface. Resolved recommendations are displayed and, if question objects exist, additional input fields

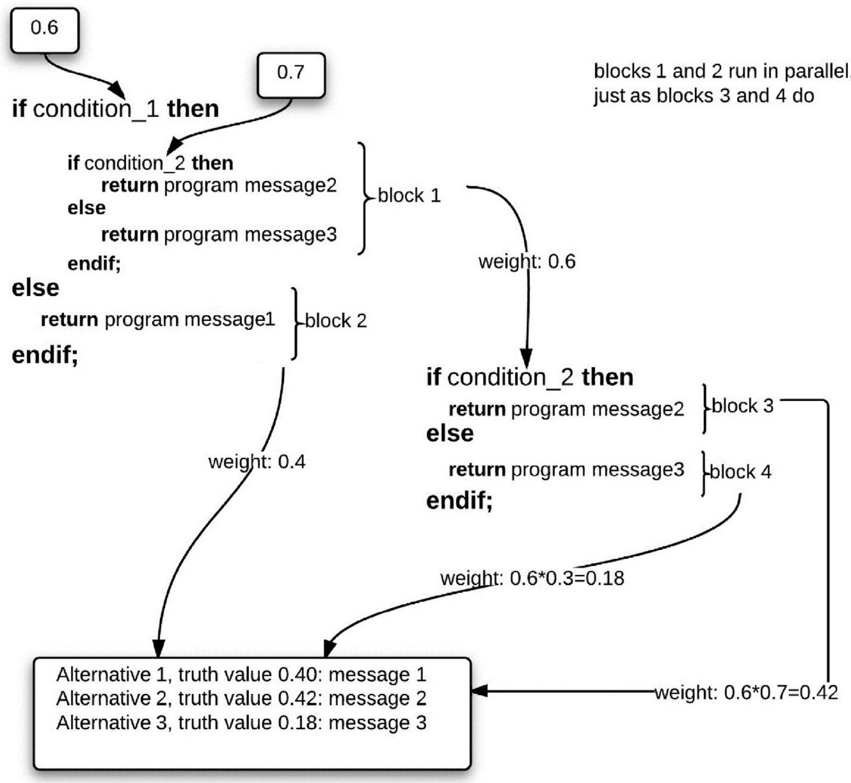

Fig. 4. This figure describes how fuzziness is computed. When evaluating an if-then-else statement of a proposition, which is neither fully true nor false but in between—presenting as a value greater than zero and less than one—the extended Fuzzy Arden Syntax engine evaluates both branches simultaneously. This is done for every if-then-else statement in an MLM, resulting in multiple final outcomes, each with its own computed degree of applicability (certainty). Operators are available for aggregating the multiple final outcomes (cf. aggregation operators in fuzzy control). Combined propositional uncertainties (through applying the fuzzy-extended operators *not*, *and*, and *or*) are calculated according to fuzzy logic.

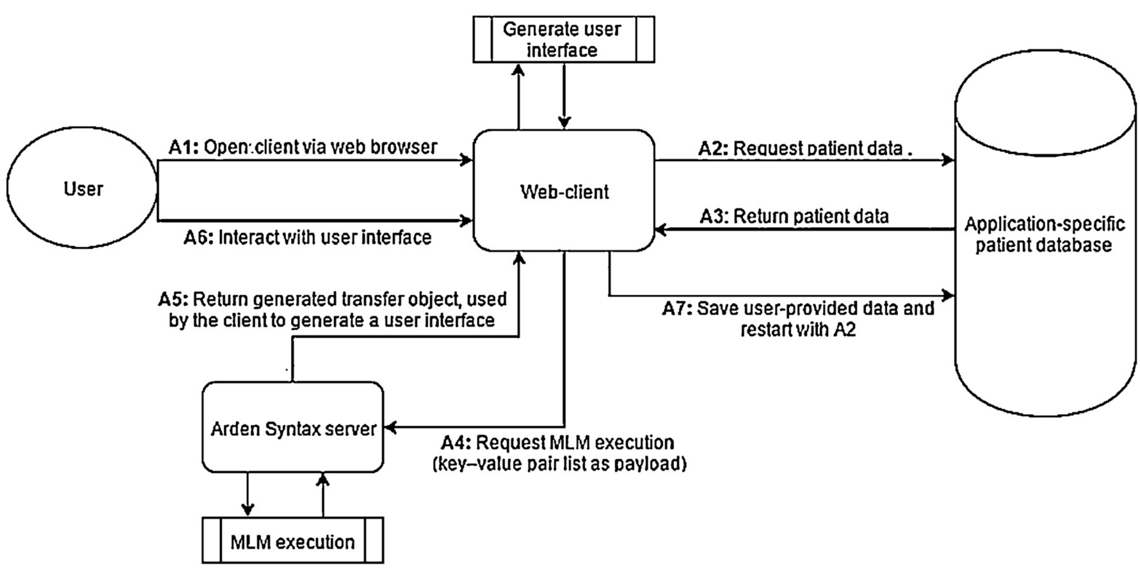

Fig. 5. The client-server communication process shown as a flowchart. After a user interface is generated for the first time, the system awaits user interaction, i.e., answering questions generated by the MLM. The entire process may be repeated. Follow action A1 to A7.

generated. Different layouts may be used in querying the question type (see Fig. 7). The CDSS prototype employs two question types: Boolean and fuzzy. While Boolean describes a question for which only the answers “yes” or “no” are allowed, fuzzy describes a question for which the user may specify the degree of certainty to which specific situations apply [41] (see Fig. 8). Using all these concepts, it is the MLM representation of a guideline that defines the client’s user interface and its elements for user interaction. Finally, if a guideline needs to be updated, MLMs are the only elements affected by these changes.

The implemented prototype uses three Lyme borreliosis guidelines, created by an Austrian expert on Lyme borreliosis [25], the German Society for Dermatology [22], and the Swiss Society for Infectious Diseases [23]. They were mapped by one of the authors

Transfer object

- **result:** A list of strings containing recommendations generated by the system.
- **next_questions:** A list of question objects to collect more patient data.
  - **q_label:** Represents the actual question as a string (text).
  - **q_type:** Describes the type of question (e.g., Boolean, fuzzy).
  - **q_info:** Additional information that would be useful to answer the question.

Fig. 6. Actual structure of the transfer object using JavaScript Object Notation, generated by an MLM, executed by the Arden Syntax server, and used to build the graphical user interface for the CDSS client, e.g., result{"Check patient for carditis"}, next_question{q_label{"Carditis"}, q_type{fuzzy}, q_info{"description of carditis"}}.

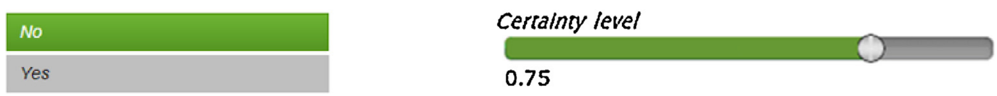

Fig. 7. Graphical user interface elements: Buttons to represent the Boolean question type, anticipating either "yes" or "no" as the answer. Sliders represent the fuzzy question type, permitting clinicians to specify the degree to which signs presented by a patient match certain symptoms of a disease. The entire real unit interval [0,1] is available, with "zero" representing false and "one" representing true.

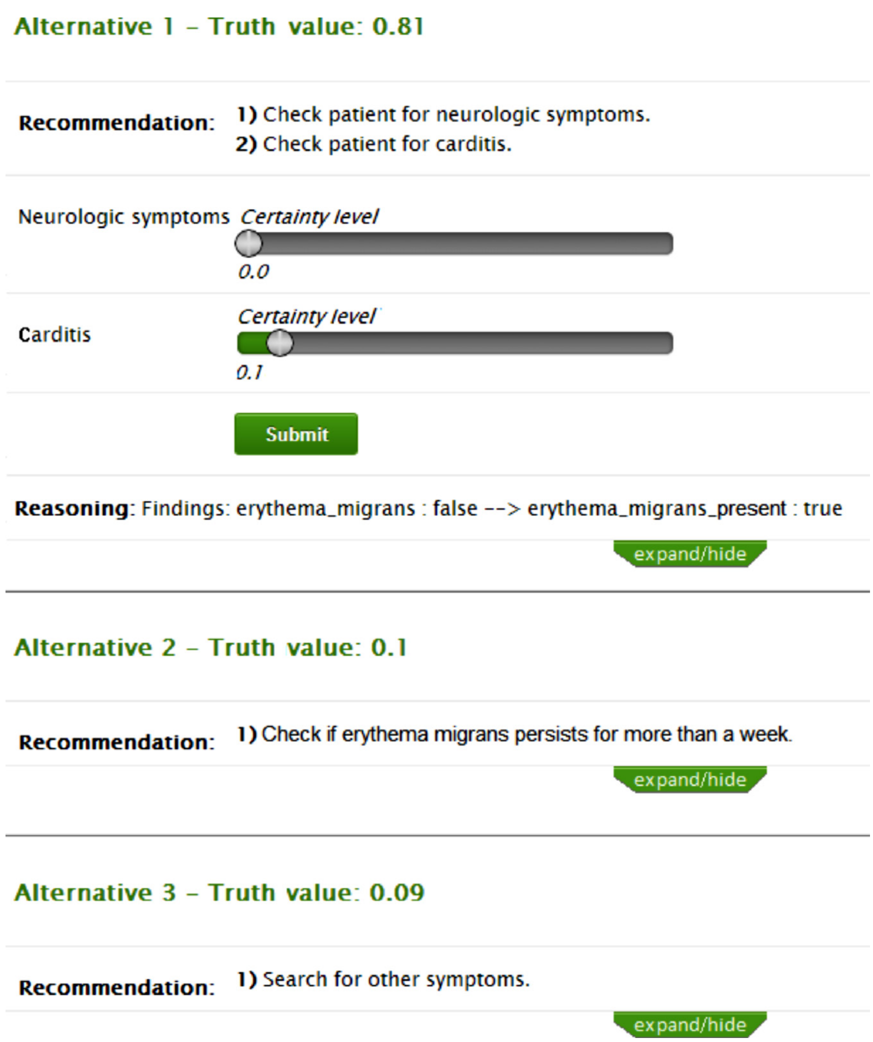

Fig. 8. A graphical user interface for the Lyme borreliosis system generated using the transfer object, which was returned by an MLM and executed by the Arden Syntax server. When a clinician provides symptoms (together with their certainty) as input, the Arden Syntax server returns recommendations as alternatives. However, each alternative is weighted according to how well it fits the provided data.

(AS) together with a dermatologist into a single combined guideline, representing the knowledge base for this project. In case of contradictions during the merging process, several medical domain experts [37] discussed the problem and agreed on a consensual definition. The final Lyme borreliosis decision tree (see Fig. 2) was used to implement corresponding MLMs. In order to provide a means of skipping irrelevant parts of a decision tree when a patient shows later-stage symptoms, predefined entry points were created. The clinician may select one during the creation of a new database patient record.

3.2. Guideline-based reminders for obstetrics

The purpose of this system [39] is to support clinicians in obstetric care by processing data according to four implemented SOPs. The results are patient-specific reminder messages in the form of to-do lists tailored to the respective phase of care: pregnancy, labor, and delivery. The system also employs a client–server architecture, using an Arden Syntax server to harbor the knowledge base in the form of SOP MLMs. While the first project uses a PHP web client, this system employs the open source Ajax Framework ZK [42] to implement its web client as a Java enterprise application, and communicates with the Arden Syntax server using a Java interface provided by the server.

The system is divided into four parts. First, the user can view all original SOP documents used as a knowledge base for further information (see Fig. 9). Second, the system allows for the management of patients' personal and medical data. This system would also benefit from EMR access, i.e., reading findings directly from the EMR's database instead of collecting data via user input. However, this solution was not implemented here. Third, a clinician may execute SOP processing. This request will generate a result report for each patient and each SOP at a given point in time (see Fig. 10). Also a general report is generated, summarizing all results of the corresponding processing time (see Fig. 11). Fourth, the generated report can be viewed and downloaded as a pdf document.

By displaying or printing reminders including only those procedures that are important for the respective patient, it is possible to prevent overlooking individual SOPs as well as incorrect or incomplete compliance due to the large number of SOPs.

4. Discussion

CDSSs have not yet become a commonplace phenomenon, although they certainly can improve the quality of clinical practice and contribute to patient safety [43]. This is partly due to systems not being well integrated into routine clinical workflows. A large number of the current CDSSs must be tightly linked to specific EMR systems in order to exploit their full potential [44]. Sharing such CDSSs with other institutions involves significant effort, resources, and software redesign. As shown here, Arden Syntax has the potential to cope with these issues.

Using the service-oriented architecture for Arden Syntax execution environments described above, institutions are able to transfer or share MLMs with each other. In cases of web-based applications, given adequate security measures, other institutions may be granted access to the implemented services (REST or SOAP) via

Group B streptococcus in pregnancy

Background:

Infection with Group B streptococcus (GBS) is the most frequent cause of neonatal sepsis. Early-onset GBS infection in neonates occurs within one to five days after birth and is the result of vertical (mother-to-child) transmission during or shortly before birth; the case fatality rate ranges between 25% and 50% in central Europe. Late-onset GBS infection in neonates occurs within five to 30 days after birth and is usually associated with meningitis; the case fatality rate ranges between 10% and 25%.

The rate of vaginal GBS colonization in pregnant women is around 5% to 25%. Vertical transmission occurs in about 50% of colonized women, and neonatal infection develops in 0.5% to 1% of these cases. The introduction of intrapartum chemoprophylaxis has reduced the frequency of neonatal GBS infection by 65%.

Diagnosis:

- The standard procedure is to obtain swabs from the vagina and the rectum (using a single cotton swab!) and to perform GBS cultures.
- Rapid diagnostic tests from vaginal secretions

Management during pregnancy:

Approach A: GBS screening using vaginal swabs between 35 to 37 weeks´ gestation during routine pregnancy care

Approach B: If no GBS screening in pregnancy has been done, intrapartum chemoprophylaxis in high-risk patients (see below)

In any case:

- Obtain vaginal swabs in all patients with threatened preterm birth (preterm labor or preterm premature rupture of the membranes – see respective SOPs)
- If GBS cultures are positive, write a clear pop-up reminder in the PIA computer program

Fig. 9. An original Streptococcus SOP used at the obstetrics department of Vienna General Hospital.

Group B streptococcus Reminder

Patient:	
Anna (model patient) Berger	(SINr. 3224120982)
First day of last menstrual period:	2014-09-09
Estimated day of delivery:	2015-06-17
Week of gestation:	38
Admitted by:	Delivery ward

Findings:	
GBS Screening:	Positive
Risk factor: Neonatal sepsis in previous pregnancy:	Negative
Risk factor: Rupture of membranes:	Negative
Risk factor: GBS test:	Negative
Risk factor: Intrapartum fever:	Negative
Risk factor: Preterm delivery:	Negative

Instructions:

- Intrapartum chemoprophylaxis with Penicillin G 10 million units intravenously every eight hours from the onset of labor until delivery
- In case of penicillin allergy: Cephalosporins (first generation) every eight hours (e.g. ZOLICEF® 2g intravenously)
- In case of cross-reactivity between penicillins and cephalosporins: Clindamycin (e.g. DALACIN® 600mg) intravenously every eight hours until delivery
- In primary caesarean sections at term, no intrapartum chemoprophylaxis is indicated in patients with positive GBS screening. However, standard antibiotic prophylaxis during caesarean section is indicated (according to SOP "Antibiotic prophylaxis in caesarean section").

Fig. 10. A generated Streptococcus SOP report, including corresponding patient data, relevant medical findings, and instructions as to what has to be done.

Internet. The ability to access web-based applications from any location via a hospital's intranet promotes continuous integration into clinical routine, making CDS available when needed.

Similar to the SANDS approach in [45], an Arden Syntax server has an architecture that provides a means of executing CDS across a network and sharing CDS content. Since Arden Syntax also has a service-oriented approach and supports modularity, it can be used in a centralized manner. The platform components are divided into front-end and back-end, and may be deployed on separate servers.

As described here, the frequently cited "curly braces problem" [6,17,31–34] grants developers the freedom to use various software design and programming patterns for external data access. For some implementations this may, but need not necessarily, result in excessively tight coupling. The problem exists in other current languages as well, although it is not as obvious in some of these. GLIF, for instance, provides a uniform data model and thus facilitates data access for developers [14]. If a uniform data model could pass data for an MLM call, this could be achieved for Arden Syntax as well. However, doing so would merely shift the data access problem to another level of abstraction: from the MLM developer to the systems developer. When uniform data models are used, institutions have to convert their medical data according to such models before being able to use them. GLIF's developers have acknowledged that the goal—to support the sharing of guidelines between different institutions and software systems—is currently impractical, given the evolving nature and apparent conceptual differences amongst the competing guideline representations [10].

Arden Syntax does not define a standard vocabulary for symptoms, laboratory test results, diagnoses, treatment procedures, and medication. As a result, CDSS developers are free to support one or several terminology standards in their CDSSs. Arden Syntax does not hinder the development of a data model that may be employed by several institutions. If a standard vocabulary is used, hospitals with different clinical terminologies have to transform their data according to the standard vocabulary, regardless of the knowledge or guideline representation language they use.

In conclusion, CDSSs should focus on the ease of medical knowledge management by not requiring adaptations to the rest of the

Summary

Patient:

Anna (model patient) Berger	(SINr. 3224120982)
First day of last menstrual period:	2014-09-09
Estimated day of delivery:	2015-06-17
Week of gestation:	38
Admitted by:	Out-patient clinic

Instructions:

- Intrapartum chemoprophylaxis with Penicillin G 10 million units intravenously every eight hours from the onset of labor until delivery
- In case of penicillin allergy: Cephalosporins (first generation) every eight hours (e.g. ZOLICEF® 2g intravenously)
- In case of cross-reactivity between penicillins and cephalosporins: Clindamycin (e.g. DALACIN® 600mg) intravenously every eight hours until delivery
- In primary caesarean sections at term no intrapartum chemoprophylaxis is indicated in patients with positive GBS screening. However, standard antibiotic prophylaxis during caesarean section is indicated (according to SOP "Antibiotic prophylaxis in caesarean section").

Gestational week 37: (37.38.39)

Blood pressure, urine dipstick, weight, home blood pressure monitoring, 24-hour blood pressure monitoring, preeclampsia education. In patients with mild hypertension/preeclampsia: Surveillance (reappointment at term). In patients with severe hypertension/preeclampsia: Hospitalization (further instructions).

Performance of Doppler sonography: Normal results: Reappointment only if necessary. Abnormal Results: Hospitalization or reappointment in up to one week.

Performance of fetal heart rate monitoring: Normal results: Reappointment in two weeks. Abnormal Results: Hospitalization.

Fig. 11. A general report summarizing the results of all SOPs, including patient data and all generated instructions as to what needs to be done.

software system. We emphasize that it is the responsibility of the software developer to design and implement CDSS architectures that are reusable for various institutions. We showed two examples of such implementations. Given these prerequisites, MLM developers will not need to focus on anything other than MLMs.

Acknowledgement

The authors are indebted to Karsten Fehre for his support in preparing and writing this manuscript.

References

[1] Greenes RA, editor. Clinical decision support: the road to broad adoption. 2nd ed. Amsterdam: Academic Press; 2014.

[2] Osheroff JA, Teich JM, Levick D, Saldana L, Velasco FT, Sittig DF, et al. Improving outcomes with clinical decision support: an implementer's guide. 2nd ed. Chicago: Healthcare Information and Management Systems Society (HIMSS); 2012.

[3] Peleg M. Computer-interpretable clinical guidelines: a methodological review. J Biomed Inform 2013;46(4):744–63.

[4] Peleg M, González-Ferrer A. Guidelines and workflow models. In: Greenes RA, editor. Clinical decision support: the road to broad adoption. Amsterdam: Academic Press; 2014. p. 435–64.

[5] Haug P, Jenders RA, Adlassnig K-P, Fehre K, Jung CY, Kim S. Health Level Seven Arden Syntax for Medical Logic Systems, Version 2.10, Health Level Seven International; 2014. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=372, last access: 21 June 2016.

[6] Samwald M, Fehre K, de Bruin J, Adlassnig K-P. The Arden Syntax standard for clinical decision support: experiences and directions. J Biomed Inform 2012;45(4):711–8.

[7] Adlassnig K-P, Rappelsberger A. Medical knowledge packages and their integration into health care information systems and the World Wide Web. In: Andersen SK, Klein GO, Schulz S, Aarts J, Mazzoleni MC, editors. eHealth beyond the horizon—get IT there. Proceedings of the 21st international congress of the European Federation for Medical Informatics (MIE 2008). Amsterdam: IOS Press; 2008. p. 121–6.

[8] Health Level Seven International, HL7 Standard, http://www.hl7.org/special/Committees/arden/products.cfm, last access: 5 March 2016.

[9] Sutton DR, Fox J. The syntax and semantics of the PRO*forma* guideline modeling language. J Am Med Inform Assoc 2003;10(5):433–43.

[10] Coiera E. Guide to health informatics. London: Arnold Hodder Headline Group; 2003.

[11] González-Ferrer A, ten Teije A, Fdez-Olivares J, Milian K. Automated generation of patient-tailored electronic care pathways by translating computer interpretable guidelines into hierarchical task networks. Artif Intell Med 2013;57:91–109.

[12] Miksch S, Shahar Y, Johnson P. Medizinische Leitlinien und Protokolle: das Asgaard/Asbru Projekt. Künstliche Intelligenz 1997;11(3):34–7 [in German].
[13] Peleg M, Boxwala AA, Bernstam E, Tu S, Greenes RA, Shortliffe EH. Sharable representation of clinical guidelines in GLIF: relationship to the Arden Syntax. J Biomed Inform 2001;34:170–81.
[14] Ohno-Machado L, Gennari JH, Murphy SN. The guideline interchange format: a model for representing guidelines. J Am Med Inform Assoc 1988;5(4):357–72.
[15] Health Level Seven International, Reference Information Model (RIM), http://www.hl7.org/implement/standards/rim.cfm, last access: 23 June 2016.
[16] Nadkarni PM. Metadata-driven software systems in biomedicine. London: Springer; 2011.
[17] Choi J, Bakken S, Lussier YA, Mendonça EA. Improving the human readability of Arden Syntax medical logic modules using a concept-oriented terminology and object-oriented programming expressions. Comput Inform Nurs: CIN 2006;24(4):220–5.
[18] Sordo M, Boxwala AA, Ogunyemi O, Greenes RA. Description and status update on GELLO: a proposed standardized object-oriented expression language for clinical decision support. Stud Health Technol Inform 2004;107:164–8.
[19] Wright A, Sittig DF. A four-phase model of the evolution of clinical decision support architectures. Int J Med Inform 2008;77(10):641–9.
[20] Medexter Healthcare, Arden Syntax, http://www.medexter.com/products/ardensuite, last access: 9 August 2016.
[21] Mygland A, Ljostad U, Fingerle B, Rupprecht T, Schmutzhard E, Steiner I. EFNS guidelines on the diagnosis and management of European Lyme neuroborreliosis. Eur J Neurol 2010;17:8–16.
[22] Hofmann H, Aberer E, Bruckbauer H, Fingerle V., Mempel M, Müllegger RR, et al., Kutane Manifestationen der Lyme Borreliose, Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften Leitlinien-Register 1(013/044) (February 2009), 1–12. [in German].
[23] Evision J, Aebi C, Francioli P, Peter O, Bassetti S, Gervaix A, et al. Abklärung und Therapie der Lyme-Borreliose bei Erwachsenen und Kindern. Empfehlung der Schweizerischen Gesellschaft für Infektiologie Teil 2: Klinik und Therapie. Schweizerische Ärztezeitung 2005;86(42):2375–84 [in German].
[24] Aberer E. Lyme borreliosis—an update. J Dtsch Dermatol Ges 2007;5(5):406–14 [in German].
[25] Prautzsch H, Von Baehr PR, Wilderich B, Bennefeld H, Berghoff W, Everth U, et al., Diagnostik und Therapie der Lyme-Borreliose – Leitlinien der Deutschen Borreliose-Gesellschaft, http://www.borreliosegesellschaft.de/Texte/Leitlinien.pdf [in German], last access: 23 June 2016.
[26] Fehre K, Adlassnig K-P, Service-oriented Arden-Syntax-based clinical decision support. in: Schreier G, Hayn D, Ammenwerth E, editors. Tagungsband der eHealth 2011 – Health Informatics meets eHealth – von der Wissenschaft zur Anwendung und zurück, Grenzen überwinden – Continuity of Care, 26.–27. Mai 2011, Wien. Wien: Österreichische Computer Gesellschaft; 2011. p. 123–8.
[27] Adlassnig K-P, Fehre K. Service-oriented Fuzzy-Arden-Syntax-based clinical decision support. Indian J Med Inform 2014;8(2):75–9.
[28] Duckett J. Beginning HTML, XHTML, CSS and JavaScript. Indianapolis: Wiley Publishing Inc.; 2010.
[29] Achour M, Betz F, Dovgal A, Lopes N, Magnusson H, Richter G, et al. PHP Handbuch; 2016. http://www.php.net/manual/de/, last access: 5 June 2016 [in German].
[30] Hripscak G. Writing Arden Syntax medical logic modules. Comput Biol Med 1994;24(5):331–63.
[31] Huff SM, Oniki TA, Coyle JF, Parker CG, Rocha RA. Ontologies, vocabularies and data models. In: Greenes RA, editor. Clinical decision support: the road to broad adoption. Amsterdam: Academic Press; 2014. p. 465–98.
[32] Jung CY, Sward KA, Haug PJ. Executing medical logic modules expressed in ArdenML using Drools. J Am Med Inform Assoc 2012;19(4):533–6.
[33] Shortliffe EH, Cimino JJ. Biomedical informatics: computer applications in health care and biomedicine. 4th edition London: Springer Science & Business Media; 2013.
[34] Sarkar IN, Chen ES, Rosenau PT, Storer MB, Anderson B, Horbar JD. Using Arden Syntax to identify registry-eligible very low birth weight neonates from the electronic health record. AMIA Annu Symp Proc 2014:1028–36.
[35] JavaScript Object Notation (JSON), http://www.json.org, last access: 5 June 2016.
[36] Vetterlein T, Mandl, H, Adlassnig K-P. Fuzzy Arden Syntax: a fuzzy programming language for medicine. Artif Intell Med 2010;49(1):1–10.
[37] Seitinger A, Fehre K, Adlassnig K-P, Rappelsberger A, Wurm E, Aberer E, et al. An Arden-Syntax-based clinical decision support framework for medical guidelines—Lyme borreliosis as an example. Stud Health Technol Inform 2014;198:125–32.
[38] Nau R, Christen H-J, Eiffert H. Lyme disease—current state of knowledge. Dtsch Arztebl Int 2009;106(5):72–82.
[39] Engel T. Leitlinienbasierte Erinnerungsmeldungen in der Geburtshilfe. Master thesis. Vienna: University of Technology; 2010 [in German].
[40] Engel T, Fehre K, Rappelsberger A, Adlassnig K-P. Qualitätssicherung in der Geburtshilfe durch leitlinienbasierte Entscheidungsunterstützung. In: Schreier G, Hayn D, Ammenwerth E, editors. Tagungsband der eHealth2011 – Health Informatics meets eHealth – von der Wissenschaft zur Anwendung und zurück, Grenzen überwinden – Continuity of Care, 26.–27. Mai 2011, Wien. Wien: Österreichische Computer Gesellschaft; 2011. p. 313–8 [in German].
[41] Kosko B. Fuzziness vs. probability. Int J Gen Syst 1990;17:211–40.
[42] Framework ZK, http://www.zkoss.org/, last access: 5 June 2016.
[43] Murphy EV. Clinical decision support: effectiveness in improving quality processes and clinical outcomes and factors that may influence success. Yale J Biol Med 2014;87:187–97.
[44] Kawamoto K, Jacobs J, Welch BM, Huser V, Paterno MD, Del Fiol G, et al. Clinical information system services and capabilities desired for scalable, standards-based, service-oriented decision support: consensus assessment of the Health Level 7 Clinical Decision Support Work Group. AMIA Annu Symp Proc 2012:446–55.
[45] Wright A, Sittig DF. SANDS: a service-oriented architecture for clinical decision support in a national health information network. J Biomed Inform 2008;41(6):962–81.