The Arden Syntax standard for clinical decision support: experiences and directions

Matthias Samwald [a,b], Karsten Fehre [c], Jeroen de Bruin [a], Klaus-Peter Adlassnig [a,c]

[a] Section for Medical Expert and Knowledge-Based Systems, Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna. Spitalgasse 23, A-1090 Vienna, Austria.

[b] Information & Software Engineering Group, Institute of Software Technology and Interactive Systems, Vienna University of Technology. Favoritenstrasse 9-11/188, A-1040 Vienna, Austria.

[c] Medexter Healthcare GmbH. Borschkegasse 7/5, A-1090 Vienna, Austria.

Corresponding author:

Dr. Matthias Samwald

E-Mail: matthias.samwald@meduniwien.ac.at

Phone: +43-699-811-68-028

Fax: +43-1-40400-6667

Address: Section for Medical Expert and Knowledge-Based Systems, Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna, Spitalgasse 23, A-1090 Vienna, Austria

Abstract

Arden Syntax is a widely recognized standard for representing clinical and scientific knowledge in an executable format. It has a history that reaches back until 1989 and is currently maintained by the Health Level 7 (HL7) organization. We created a production-ready development environment, compiler, rule engine and application server for Arden Syntax. Over the course of several years, we have applied this Arden – Syntax – based CDS system in a wide variety of clinical problem domains, such as hepatitis serology interpretation, monitoring of nosocomial infections or the prediction of metastatic events in melanoma patients. We found the Arden Syntax standard to be very suitable for the practical implementation of CDS systems. Among the advantages of Arden Syntax are its status as an actively developed HL7 standard, the readability of the syntax, and various syntactic features such as flexible list handling. A major challenge we encountered was the technical integration of our CDS systems in existing, heterogeneous health information systems. To address this issue, we are currently working on incorporating the HL7 standard GELLO, which provides a standardized interface and query language for accessing data in health information systems. We hope that these planned extensions of the Arden Syntax might eventually help in realizing the vision of a global, interoperable and shared library of clinical decision support knowledge.

Keywords: Arden Syntax, clinical decision support, standards, HL7

1. Introduction

Decision making in modern medical practice is based on increasingly complex medical knowledge and clinical evidence. This makes it difficult to provide the best possible care in the busy environment typically encountered in healthcare settings. It has been shown that clinical decision support (CDS) systems can significantly improve the quality of treatment if they meet certain design criteria [1].

Arden Syntax is a widely recognized standard for representing clinical and scientific knowledge in an executable format which can be used by such CDS systems. It has a long history: A first draft of the standard was prepared at a meeting at the Arden Homestead, New York, in 1989. The first Arden Syntax specification was published by the American Society for Testing and Materials (ASTM) in the year 1992. Later on, the standard was integrated into Health Level 7 (HL7) [2]. HL7 published Arden Syntax version 2.0 in the year 1999 and has been hosting the development of all newer versions of the Arden Syntax standard ever since. The Arden Syntax Working Group [3] is currently responsible for the active development of the standard within HL7. The present, most recent version of Arden Syntax is version 2.8.

Any common programming language can, in theory, be used to implement CDS systems. However, the Arden Syntax standard was designed for this specific purpose and is equipped with a set of features that make it especially useful for this task. Arden Syntax can be used in a way that makes program code resemble natural language, which, in turn, makes the code easier to understand by non-experts in computer science. It also features a choice of data types that is tailored to the needs of medical documentation, including measures for time and duration. Additionally, the use of Arden Syntax makes it possible to represent decision support logic independently from the programming languages and implementation details chosen for a specific hospital information system (HIS) , making it easier to exchange CDS logic between different systems at different sites.

Arden Syntax can be seen as a hybrid between classical production rules and procedural representation of clinical algorithms. The code is organized in self-contained files called Medical Logic Modules (MLMs). The execution of an MLM can be triggered by specific data- or time-based events or by a direct call. An example of a simple MLM is shown in Fig. 1. In this paper we provide a brief overview of the basic features of Arden Syntax; a more detailed description can be found in [4].

```
maintenance:
        title: simple body mass index;;
        mlmname: BMI;;
        arden: Version 2.8;;
        version: 1.00;;
        institution: Medexter Healthcare;;
        author: Karsten Fehre;;
        specialist: ;;
        date: 2011-11-01;;
        validation: testing;;
library:
        purpose: body mass index;;
        explanation:
                calculation of body mass index (BMI)
                input: compound list with:
                        (number) size in m,
                        (number) weight in kg,
                        (time) birth date.
                output:
                        If the age is not less than 19 and the classification
                        according to WHO is not normal, a message containing
                        the calculated BMI and its classification will be sent.
;;
        keywords: BMI, body mass index;;
        citations: ;;
        links: http://en.wikipedia.org/wiki/Body_mass_index;;
knowledge:
        type: data_driven;;
        data:
```

```
            // Arguments
            (size, weight, birth) := argument;

        ;;
        priority: ;;
        evoke: ;;
        logic:
          // calculation of BMI
          let bmi be weight / (size ** 2); // BMI

          // calculation of AGE
          age := currenttime - birth; // AGE

          // classification
          if the age is less than 19 years then classification := null;
          // This classification is only valid for patients older than 19

          elseif the bmi is less than 18.5 then classification := localized 'under';
          // classified as underweight

          elseif the bmi is less than 25 then classification := null; // BMI normal range

          else let the classification be localized 'over';
           // classified as overweight
          endif;

          bmi := bmi formatted with localized 'msg'; // construct the localized message

          conclude classification is present ; // if there is a classification, execute
           the action slot
        ;;
        action:
          write bmi || classification || "."; // return result
          return classification;
        ;;
        urgency: ;;

resources:
        default: de;;
        language: en
            'msg' : "The patient's BMI %.1f is not in the normal range and is classified as ";
```

```
            'under' : "underweight.";
            'over' : "overweight."
    ;;
    language: de
            'msg' : "Der BMI %.1f des Patienten ist nicht im normalen Bereich und wird
    klassifiziert als ";
            'under' : "Untergewicht.";
            'over' : "Übergewicht."
    ;;
end:
```

Figure 1: An example of an MLM in Arden Syntax. It calculates the body mass index (BMI) given the parameters size, weight and birth date.

All MLMs follow a rigid structure to ensure that processing code representing medical knowledge and program logic, such as formal rules derived from clinical guidelines, is kept separate from more technical code, such as variable declarations and interfacing with external data sources and services. This division helps to improve the transparency of the MLM code.

The content of an MLM is divided into three 'categories', which in turn are divided into 'slots'. The first category is the *maintenance* category. It contains metadata about the MLM, e.g., its author and the creation date. This is followed by the *library* category, which contains background information, e.g., an explanation of what the MLM is supposed to accomplish and references to relevant medical literature. The *knowledge* category contains the actual algorithm for decision making and also defines database access mechanisms and events. It is divided into several slots to improve transparency. For example, the *knowledge->evoke* slot defines under which circumstances (events) an MLM is to be triggered. When these criteria are met, the MLM may start immediately, after a specific waiting time, or even periodically. The *knowledge->logic* slot contains the logic and procedures for decision making based on input data. Finally, the *knowledge->action* slot writes the result to the screen or a database, sends an e-mail, or calls other MLMs for further processing.

The fact that one MLM can invoke another MLM makes it possible to modularize complex decision making processes by distributing them over several re-usable MLMs.

The Arden Syntax standard is still evolving. For example, the release of Arden Syntax version 2.8 in January 2012 brought the addition of more powerful list operators and more control flow operators. The release of Arden Syntax version 2.9 (expected for the May 2012 ballot cycle) is planned to bring the addition of gradual truth values, fuzzy sets and fuzzy logic operators as well as an expanded XML representation of Arden Syntax code.

Examples of systems that implement the Arden Syntax standard include the CARE system and related implementations developed by the Regenstrief Institute (such as the system described in [5]), an Arden Syntax rule engine integration into the Soarian system offered by Siemens, the *ORBIS Experter* product offered by Agfa Healthcare , the *Horizon Care Alerts* computerized nursing documentation system offered by McKesson [6], the software products offered by Allscripts [7], and the software products offered by Medexter Healthcare [8].

Besides these closed-source products, there are also two documented open-source implementations of Arden Syntax compilers: Arden/J [9] and Arden2bytecode [10]. However, the completeness and usability of these implementations could not be evaluated.

In this paper we report our experience with developing, deploying, and using CDS systems based on Arden Syntax and its extended version, Fuzzy Arden Syntax, over the course of several years. The work reported here was done collaboratively by the Section for Medical Expert and Knowledge-Based Systems at the Medical University of Vienna and the Medexter Healthcare company.

2. Material and methods

We mainly relied on software solutions developed in collaboration with Medexter Healthcare. The company maintains an up-to-date implementation of an Arden Syntax compiler and rule engine, a dedicated server for interfacing with existing health care systems, and an integrated Arden Syntax development and test environment (IDE).

We used the *Arden Syntax IDE* to develop MLMs. This software provides a simple development environment for Arden Syntax code, including syntax highlighting and testing MLMs. MLMs were compiled using the *Arden Syntax compiler*, which generates Java classes out of MLM code. The *Arden Syntax rule engine* was used to execute these compiled MLMs. The rule engine can be seen as a specialized virtual machine or player for compiled MLMs. It operates in concert with another component called the *MLM manager,* which gives the Arden Syntax rule engine programmatic access to all available

MLMs in the system. The *Arden Syntax server* was used to provide system/language independent access to the rule engine [11]. For projects that relied on fuzzy logic, we used the enhanced *Fuzzy Arden Syntax IDE* and *Fuzzy Arden Syntax rule engine* [12,13]. All of these software products were implemented in Java.

3. Results and Discussion

Below we describe some of the Arden Syntax implementations we worked on over the past years. Then we summarize the limitations and advantages of the Arden Syntax standard that we encountered during our work. We pay special attention to considerations dealing with system integration and semantic interoperability, as these are of major importance for the successful adoption of the standard.

3.1 Implementations

Our group developed many different clinical decision support systems using the Arden Syntax standard. They range from basic systems that contain only few and simple MLMs to quite sophisticated systems composed of many dozens of MLMs.

Hepaxpert, Thyrexpert, Toxopert, RHEUMexpert

The *Hepaxpert* system assists in the interpretation and plausibility checking of hepatitis A, B, and C serology test results [14,15]. As with many others of our decision support tools, it can be used both through a web interface (exemplified in Fig. 2 and Fig. 3) as well as an API. The Hepaxpert system is made up of a small number of MLMs: a main MLM that evokes the other MLMs in the collection, three MLMs containing decision making logic, and one MLM containing output text templates. The domain knowledge contained in Hepaxpert was gathered over the course of several years through collaborations with clinicians. The textual representations are built by concatenating static output text templates with the values observed in individual patients. Several dozens of these text templates are integrated into the Hepaxpert system to cover a wide variety of possible outcomes in detail. The MLMs can be managed and updated through a management interface.

Hepaxpert has been successfully integrated with the hospital information systems Soarian (offered by Siemens) and Orbis (offered by Agfa Healthcare). Additionally, an interface for the Hepaxpert tool has been developed for the iPhone and iPad and will be available through the Apple App Store soon.

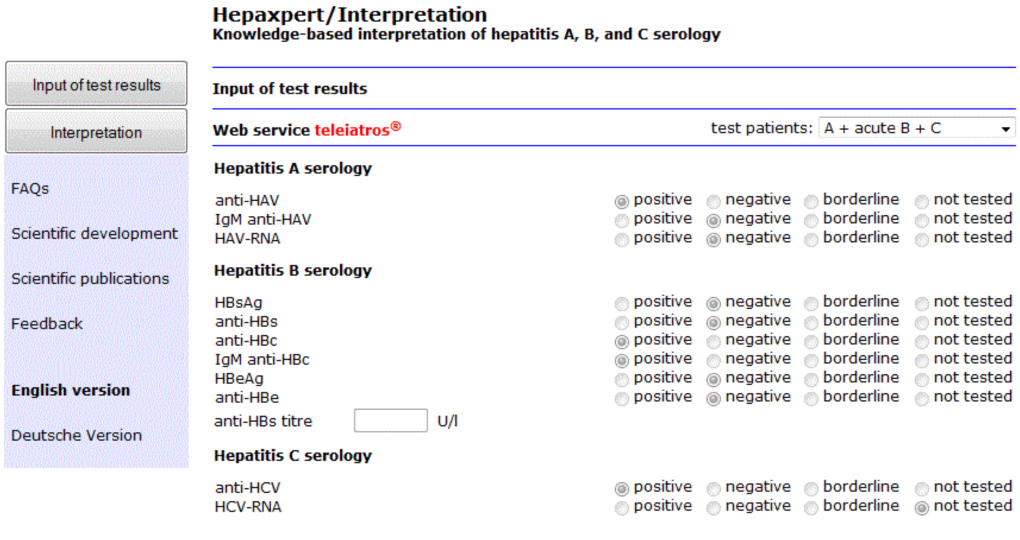
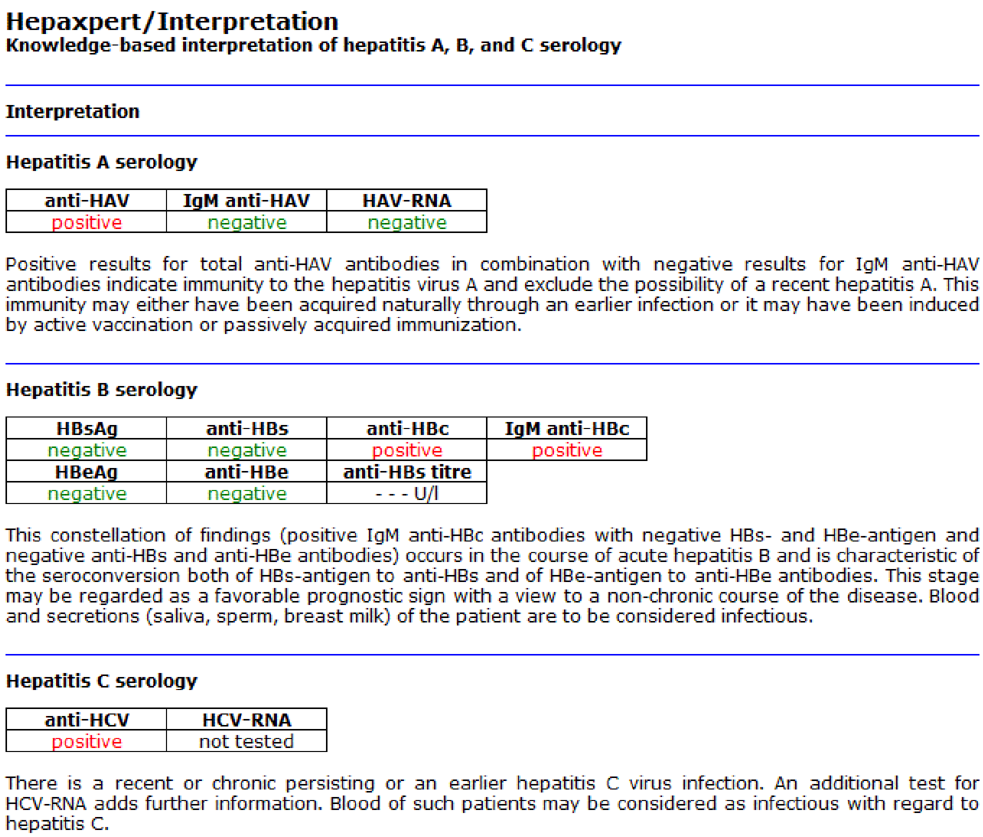


Figure 2: Hepaxpert data entry in a web interface.



Hepaxpert/Interpretation
Knowledge-based interpretation of hepatitis A, B, and C serology

Interpretation

Hepatitis A serology

anti-HAV	IgM anti-HAV	HAV-RNA
positive	negative	negative

Positive results for total anti-HAV antibodies in combination with negative results for IgM anti-HAV antibodies indicate immunity to the hepatitis virus A and exclude the possibility of a recent hepatitis A. This immunity may either have been acquired naturally through an earlier infection or it may have been induced by active vaccination or passively acquired immunization.

Hepatitis B serology

HBsAg	anti-HBs	anti-HBc	IgM anti-HBc
negative	negative	positive	positive
HBeAg	**anti-HBe**	**anti-HBs titre**	
negative	negative	- - - U/l	

This constellation of findings (positive IgM anti-HBc antibodies with negative HBs- and HBe-antigen and negative anti-HBs and anti-HBe antibodies) occurs in the course of acute hepatitis B and is characteristic of the seroconversion both of HBs-antigen to anti-HBs and of HBe-antigen to anti-HBe antibodies. This stage may be regarded as a favorable prognostic sign with a view to a non-chronic course of the disease. Blood and secretions (saliva, sperm, breast milk) of the patient are to be considered infectious.

Hepatitis C serology

anti-HCV	HCV-RNA
positive	not tested

There is a recent or chronic persisting or an earlier hepatitis C virus infection. An additional test for HCV-RNA adds further information. Blood of such patients may be considered as infectious with regard to hepatitis C.

Figure 3: Hepaxpert interpretation result in a web interface.

Thyrexpert assists in the interpretation of thyroid hormone test results and offers automated reminders for quality assurance of thyroid diagnoses [16,17]. *Toxopert* assist in the interpretation of time sequences of toxoplasmosis serology test results [18,19]. This application is also available in an enhanced version that is based on fuzzy logic [19]. *RheumaDiff* and *RHEUMexpert* offer differential diagnostic decision support in rheumatology [20,21]. The technical implementation and user interfaces of Thyrexpert, Toxopert, RheumaDiff and RHEUMexpert resemble those of Hepaxpert.

Prediction of metastatic events in melanoma patients

In a joint project with the Department of Dermatology at the Vienna General Hospital and the Siemens Austria company a service-oriented CDS system to predict metastatic events in melanoma patients [22] was established. In this implementation, the Arden Syntax server is communicating with the Vienna General Hospital's information system through a service interface, and the results generated by the CDS system are seamlessly integrated into the user interface of the HIS. The service-oriented architecture makes it possible to host the CDS system on a different server than the main HIS. Such a division can be preferable in many cases, since the addition of new software on servers hosting the HIS could potentially impact the security and stability of the main HIS system. When CDS algorithms are hosted on dedicated servers and communication with the main HIS is restricted to a bare minimum of necessary service calls, these concerns can be reduced.

The system is composed of four MLMs, with one MLM containing the main decision logic and the other MLMs containing functions for specific calculations. The system is maintained by several domain experts in dermatology in cooperation with an expert in medical informatics who helps with translating the expert's knowledge into MLM code.

Moni-ICU

Moni-ICU ('monitoring of nosocomial infections in intensive care units') [23,24] is a system which detects and continuously monitors nosocomial (i.e., hospital-acquired) infections. Moni-ICU uses a knowledge base of hospital-acquired infection detection rules implemented in a modified version of the Arden Syntax language which supports fuzzy logic structures, as described in a classic paper by Zadeh [25]. The motivation for using fuzzy logic in the knowledge base is to provide physicians with more accurate information on the degree of presence of hospital-acquired infections, which helps to identify borderline cases and enables earlier detection of an infection onset and its decline [26].

The Moni-ICU application is used by the Clinical Institute of Hospital Hygiene of the Vienna General Hospital on a daily basis to monitor all patients in each of the normal intensive care unit wards, which comprise of around 100 beds in total. Moni-ICU is connected to two sources of patient data: the microbiology laboratory and the patient data management system of the ICUs. All relevant patient data from these sources is downloaded and processed on an instance of the Arden Syntax server. After processing, the attending physician can view the results as well as detailed explanations through a web interface.

The domain knowledge encoded in the Moni-ICU system primarily consists of the criteria for the presence of a nosocomial infection based on certain clinical and laboratory data. Two important sources of such criteria used in the MONI system are: 1) the *Hospitals in Europe Link for Infection Control through Surveillance* (HELICS) criteria [27] and 2) the *Krankenhaus-Infektions-Surveillance-System* (KISS, German for 'hospital infection surveillance system') criteria [28] of the Robert Koch Institute. Although KISS criteria are older than the HELICS criteria, they are a useful addition, because KISS is often employed in German healthcare institutions. Domain experts from the Vienna General Hospital assisted with implementing the criteria from these sources and provided additional expert knowledge.

The use of Arden Syntax in the Moni-ICU system is distinct from other systems presented in the paper, in that MLMs are used to create a production system. One central MLM invokes a large number of other MLMs, each implementing a single rule. The Moni-ICU interface allows doctors to see a filtered list of clinically relevant results of these production rules, including information on the patient data that was used as evidence for each conclusion.

Our group is actively involved in integrating the formalisms for representing fuzzy logic that we developed [29] into one of the upcoming next release of the Arden Syntax standard. We are currently designing a study for comparing the effectiveness of fuzzy logic with the effectiveness of crisp logic in the Moni-ICU system.

VistA health information system integration

In collaboration with the Veterans Health Administration in the United States, our group is currently working on integrating the Arden Syntax rule engine with the VistA health information system [30]. This system is in use at Veteran Affairs (VA) hospitals in the US, and is one of the most widely used health information systems in the world. In the context of this project, we are working on connecting the HL7 standard GELLO (a loose acronym for Guideline Expression Language Object-Oriented) [31] to Arden

Syntax. The connection with GELLO makes it possible to access medical data in the hospital information system in a standardized way and has the potential to significantly improve the syntactic and semantic interoperability of MLMs across institutions.

The MLMs in the VA project are based on existing, proprietary reminder applications that are already in use in some VA hospitals. Several reminders for medical acts such as eye exams, mammograms or influenza vaccination have been implemented, with each MLM representing one type of reminder. The development of a patient report card application is also under way within this context.

Other implementations

Our group also implemented prototypes for clinical decision support in obstetrics [32], planning of immunosuppressive therapy in patients that underwent a renal transplantation [33] and prenatal screening of toxoplasmosis [34]. Besides the systems intended for defined medical applications, our group developed several hundred MLMs for testing the Arden Syntax compiler and related software for standards compliance.

3.2 Limitations and advantages

Generally, we did not encounter many limitations of the Arden Syntax standard, which might be explained by the fact that the aims of our developments are well aligned with the intended uses of the standard. Some notable shortcomings:

- The Arden Syntax specification could have been written in a more clear and accessible style. Ideally, the standard would be described in two separate, focused documents, one for developers of Arden Syntax compilers, and one for users/programmers that want to develop or administer Arden Syntax MLMs.
- Different categories allow the use of different operators and statements, which can lead to inconsistencies.

We also identified some distinct advantages of the standard, for example:

- It allows for flexible handling of lists, which can contain mixtures of data types and can be filtered with ease.
- A feature of substantial practical advantage is the handling of temporal metadata, since each data item in Arden Syntax has two components: 1) the value, 2) the primary time, referring to

the time point in which the data item was gained or added to the database. This makes it easier to handle patient data generated at different times.

- The standard encourages developers to document and annotate MLMs with rich metadata, which is important for making large collections of MLMs manageable.
- It allows for dynamic typing of variables, which makes it easier to rapidly develop new code and makes the standard easier to understand for non-experts in computer science.

One major potential advantage of the Arden Syntax standard is the possibility to write code in a way that is close to natural language, in order to improve readability by domain experts (such as clinicians). For projects in which Arden Syntax code was only created and maintained within our group, we preferred to use a succinct coding style, resembling other established programming languages. For projects in which the code was shared and needed to be understood by non-experts in computer science, we used the more verbose coding style that was closer to natural language. Based on our experiences, the best way to create MLMs is to engage domain experts in an active dialog, starting out with drafting workflows and criteria with pen and paper. When a thorough consensus about the algorithm was reached, we translated it into MLM code. While domain experts without prior experience with Arden Syntax are not able to work with Arden Syntax code, we found that long-term users with a medical background can learn to modify and comment on Arden Syntax code without assistance.

Last but not least, the status of Arden Syntax as an official standard backed by the HL7 community can also be seen as an advantage over potential alternatives that have not been standardized. The status as a standard and the clearly defined community process of HL7 working groups makes it likely that Arden Syntax will be maintained and adapted to the changing needs of medical information technology systems by an international group of stakeholders for many years to come.

3.3 System integration and semantic interoperability

A CDS system can only work in practice when it is well-integrated with existing medical information systems, and non-existent or insufficient integration can be a major obstacle to the success of CDS systems. We have considered and implemented several different models of embedding CDS systems based on Arden Syntax in existing health informatics systems. The simplest approach is, of course, to use a hospital information system that has built-in support for decision support based on Arden Syntax, such as some of the products offered by Agfa or Siemens. However, in many health care settings such a system is not in use, and alternative approaches need to be considered. In these settings, we deployed CDS systems as services running on dedicated machines communicating with the medical information system through customized interfaces, and found this approach to be very effective [11].

The Arden Syntax server that we developed and used is based on what we call the *Arden Syntax host interface*, a Java interface which defines the communication between the Arden Syntax engine and the Arden Syntax server. The interface is modified to fit the needs of any particular installation.

In addition to the Arden Syntax host interface, we defined the *Arden Syntax server protocol*, which is a simple XML schema describing possible input and output messages of the server. This server protocol is not tailored towards specific implementations and can be used as a universal interface to the server.

While these interfaces help to make the communication between the Arden-Syntax-based CDS and the medical information system more transparent, they do not solve the fundamental problem of semantic interoperability. Ideally, it should be possible to re-use a specific MLM (such as an algorithm for determining the body mass index) at many institutions without the need to adapt the MLM for each institution. However, different medical information systems at different locations use different data structures, identifiers and coding schemes. This means that work needs to be invested in customization for each new installation, be it the adaptation of interfaces in the medical information system, the Arden Syntax server, or the code in the MLM itself.

This problem is often referred to as the 'curly braces problem' in the Arden Syntax community. In an MLM, curly braces are used to demarcate those parts of the code that are specific to the surrounding health information system (such as routines for accessing data from the patient database at a specific site). Since these code fragments in the MLM need to be adapted for each specific installation, it is very difficult to create a general repository of MLMs and knowledge bases that could be shared between institutions.

Some proposals for solving or at least ameliorating the curly braces problem have been published (e.g., [35]). Recent developments converge to a unified solution that will become part of the Arden Syntax standard, probably within the next three years. One possible solution is based on integrating the standard expression language GELLO [31,36] into MLMs for communicating with medical information systems through a standardized interface, the virtual medical record (vMR). As mentioned above, we are working on such integration in the context of our collaboration with the Veterans Health Association. An example of an Arden Syntax code snippet that shows how GELLO and vMR access inside Arden

Syntax looks like is presented in Fig. 4. Current solutions for integration with host systems (such as the Arden Syntax host interface) will largely become obsolete once the integration of GELLO expressions into Arden Syntax is finalized.

```
data:
    // 1. define an interface which is called to get the birthday of a person
    LET get_birth BE INTERFACE {Patient.dateOfBirth};

    // 2. assume the patient ID is passed to the MLM
    LET patientID BE argument;

    // 3. call the interface with the passed patient ID
    LET birth BE CALL get_birth WITH patientID;
```

Figure 4: An example of embedding GELLO expressions in Arden Syntax code to access a virtual medical record (vMR).

3.4 Related standards, formats and architectures

Based on the work of Kawamoto *et al.* [37] we can conceptualize Arden Syntax as being one element in a system of interrelated standards that can be used to realize computational CDS. The major target areas of standardization efforts are:

- Terminologies (e.g., RxNorm [38])
- Information models (e.g., HL7 version 2 and 3 information models)
- Patient data expected to be available for CDS (e.g., HL7 vMR standard [39])
- Approaches for terminology and ontology inferencing (e.g., the Web Ontology Language / OWL [40], HL7 Common Terminology Services standard [41])
- Representation of clinical knowledge in non-executable format suitable for translation into executable format (e.g., ASTM International Guideline Elements model [42])

- Representation of clinical knowledge in an executable format (e.g., the Arden Syntax standard discussed in this paper)
- Approaches to utilizing machine-executable clinical knowledge to generate CDS and to interact with health information systems (e.g., HL7 Decision Support Service [43])

By aligning with these related standardization efforts, the utility of Arden Syntax might be increased further. As described, we are already working on connecting GELLO / vMR to Arden Syntax. The alignment with standardized terminologies and service interfaces (e.g. based on the HL7 Decision Support Service specification) might prove useful as well.

Within the area of executable clinical knowledge representation, there are some standards, formats and architectures that can be seen as possible alternatives to the Arden Syntax standard. Among them are specialized representation languages for clinical guidelines such as the *Guideline Interchange Format* (GLIF) [44]. These differ from Arden Syntax in that they are more focused on complex clinical workflows and decision trees. Furthermore, GLIF requires the use of dedicated graphical development environments, since the code used to describe guidelines is not quite human-readable. The complexity of GLIF might cause an unwanted overhead when the goal of CDS is to generate simple alerts and reminders [45]. To date, GLIF was not published as a standard by a standards development organization. The Rule Interchange Format (RIF) [46] has recently become a recommendation of the World Wide Web consortium (W3C). It is intended to serve as a common format for representing and exchanging general rules, and could therefore also be used to represent medical decision support rules. At the time of this writing, there are few implementations of tools based on RIF and no medical application has been documented.

Wright & Sittig [47] propose a framework for comparing clinical decision support architectures and provide a comparison of Arden Syntax with some alternative approaches. They conclude that all approaches, including Arden Syntax, have specific advantages and trade-offs that need to be considered for each application area. According to their analysis, some possible shortcomings of Arden Syntax as compared to other approaches are the lack of standardized vocabularies and patient data schemas, lacking integration in service-oriented architectures and difficulties with providing clinicians with actionable choices that influence program execution. We have addressed some of these issues in our work.

In our experience, among the biggest advantages of Arden Syntax are its status as an established HL7 standard backed by an international community, the availability of production-ready commercial tools, and its flexible and easily readable syntax.

3.5 Future work

There are several ways in which the adoption and positive impact of Arden Syntax and surrounding tools can be increased. A major obstacle for new users of the standard is the lack of good, up-to-date tutorials and end-user manuals. To remedy this situation, we are currently working on writing and publishing introductory reading material so that new users can quickly learn the basics of Arden Syntax (and soon of Fuzzy Arden Syntax).

A goal for future work by the Arden Syntax community as a whole is the creation of reference MLMs that can be used by developers of Arden Syntax compilers to test compliance with the standard. We have already accomplished some ground work in this area by developing a large library of test MLMs.

Finally, improvements in future versions of Arden Syntax, such as the reliance on GELLO and vMRs for improved interoperability, open up the possibility of creating shared libraries of medical logic in the form of Arden Syntax MLMs. We can envision that such libraries will be both available in open access repositories and as commercial products. Similar to the popular 'App Store' maintained by the computer company Apple, an 'MLM Store' could offer validated, interoperable MLMs, enabling the development of a lively market place for CDS software.

4. Conclusions

The Arden Syntax standard has now a history of about two decades. While problems with syntactic and semantic interoperability with heterogeneous health information systems had a negative impact on widespread adoption of this standard in the past, recent developments of the standard are bound to minimize these problems. These developments are likely to increase the utilization of the standard, and might revive the original ideals driving its development: the free and unencumbered creation and sharing of CDS logic for improving clinical practice and, ultimately, the quality of life of patients.

Conflicts of interest

Klaus-Peter Adlassnig is also co-founder, CEO and scientific head of Medexter Healthcare, established to broadly disseminate decision support systems with clinically proven usefulness.

References

[1] K. Kawamoto, Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success, BMJ. 330 (2005) 765-0.

[2] Health Level Seven International - Homepage. Available: http://www.hl7.org/ (last accessed: January 31, 2012)

[3] Arden Syntax. Available: http://www.hl7.org/special/Committees/arden/index.cfm (last accessed: January 31, 2012)

[4] G. Hripcsak, Writing Arden Syntax medical logic modules, Computers in Biology and Medicine. 24 (1994) 331-363.

[5] V. Anand, P.G. Biondich, G. Liu, M. Rosenman, S.M. Downs, Child Health Improvement through Computer Automation: the CHICA system, Stud Health Technol Inform. 107 (2004) 187-191.

[6] Computerized Nursing Documentation Systems Improves Health Care Quality | Horizon Care Alerts | McKesson. Available: http://www.mckesson.com/en_us/McKesson.com/For%2BHealthcare%2BProviders/Hospitals/Interdisciplinary%2BCare%2BSolutions/Horizon%2BCare%2BAlerts.html (last accessed: January 31, 2012)

[7] Allscripts - Home. Available: http://www.allscripts.com/ (last accessed: January 31, 2012)

[8] Medexter Healthcare Home. Available: http://www.medexter.com/ (last accessed: January 31, 2012)

[9] H.C. Karadimas, C. Chailloleau, F. Hemery, J. Simonnet, E. Lepage, Arden/J: An Architecture for MLM Execution on the Java Platform, Journal of the American Medical Informatics Association. 9 (2002) 359 -368.

[10] dgrunwald/arden2bytecode - GitHub. Available: https://github.com/dgrunwald/arden2bytecode (last accessed: January 31, 2012)

[11] K. Fehre, K.-P. Adlassnig, Service-Oriented Arden-Syntax-Based Clinical Decision Support, in: Proceedings of eHealth2011, Austrian Computer Society, Vienna, 2011: S. 123 - 128.

[12] T. Vetterlein, H. Mandl, K.-P. Adlassnig, Fuzzy Arden Syntax: A fuzzy programming language for medicine, Artificial Intelligence in Medicine. 49 (2010) 1-10.

[13] K. Fehre, H. Mandl, K.-P. Adlassnig, A fuzzy Arden Syntax compiler, in: Proceedings of eHealth2010, Austrian Computer Society, Vienna, o. J.

[14] Medexter Healthcare - Hepaxpert. Available: http://www.medexter.com/Hepaxpert/ (last accessed: January 31, 2012)

[15] C. Chizzali-Bonfadin, K.P. Adlassnig, M. Kreihsl, A. Hatvan, W. Horak, A WWW-accessible knowledge base for the interpretation of hepatitis serologic tests, Int J Med Inform. 47 (1997) 57-60.

[16] M. Reitstatter, H. Vierhapper, A. Rappelsberger, K.-P. Adlassnig, Quality Improvement through a Reminder System in an Outpatient Department for Thyroid Disease, in: Computational Intelligence for Modelling, Control and Automation, International Conference on, IEEE Computer Society, Los Alamitos, CA, USA, 2006: S. 142.

[17] Medexter Healthcare - Thyrexpert. Available: http://www.medexter.com/Thyrexpert/ (last accessed: January 31, 2012)

[18] S. Nagy, M. Hayde, B. Panzenböck, K.P. Adlassnig, A. Pollak, Toxopert-I: knowledge-based automatic interpretation of serological tests for toxoplasmosis, Comput Methods Programs Biomed. 53 (1997) 119-133.

[19] Medexter Healthcare - Toxopert, http://www.medexter.com/Toxopert/InterpretationFuzzy/index.php (last accessed: January 31, 2012)
[20] Medexter Healthcare Rheumadiff. Available: http://www.medexter.com/index.php?stateRequest=rheumadiff (last accessed: January 31, 2012)
[21] G. Kolarz, K.P. Adlassnig, K. Bögl, [RHEUMexpert: a documentation and expert system for rheumatic diseases], Wien Med Wochenschr. 149 (1999) 572-574.
[22] T. Mehl, M. Binder, C. Scheibboeck, S. Holub, K.-P. Adlassnig, Integration of Clinical Decision Support into a Hospital Information System to Predict Metastatic Events in Patients with Melanoma, in: Proceedings of eHealth2010, Austrian Computer Society, Vienna, o. J. S. 107 - 112.
[23] K.-P. Adlassnig, A. Blacky, W. Koller, Artificial-intelligence-based hospital-acquired infection control, Stud Health Technol Inform. 149 (2009) 103-110.
[24] A. Blacky, H. Mandl, K.-P. Adlassnig, W. Koller, Fully Automated Surveillance of Healthcare-Associated Infections with MONI-ICU, Applied Clinical Informatics. 2 (2011) 365-372.
[25] L.A. Zadeh, Fuzzy sets, Information and Control. 8 (1965) 338-353.
[26] W. Adlassnig, K.-P., Blacky, A., Koller, Fuzzy-based nosocomial infection control, Studies in Fuzziness and Soft Computing. 218 (2008) 343-349.
[27] C. Suetens, A. Savey, J. Labeeuw, I. Morales, The ICU-HELICS programme: towards European surveillance of hospital-acquired infections in intensive care units, Euro Surveill. 7 (2002) 127-128.
[28] P. Gastmeier, C. Geffers, J. Koch, D. Sohr, A. Nassauer, F. Daschner, u. a., Surveillance nosokomialer Infektionen: Das Krankenhaus-Infektions-Surveillance-System (KISS)., LaboratoriumsMedizin. 23 (1999) 173-178.
[29] T. Vetterlein, H. Mandl, K.P. Adlassnig, Processing gradual information with Fuzzy Arden syntax, Stud Health Technol Inform. 160 (2010) 831-835.
[30] Enhanced CPRS Clinical Decision Support. Available: http://sites.google.com/site/enhancedcprscds/home (last accessed: January 31, 2012)
[31] M. Sordo, A.A. Boxwala, O. Ogunyemi, R.A. Greenes, Description and status update on GELLO: a proposed standardized object-oriented expression language for clinical decision support, Stud Health Technol Inform. 107 (2004) 164-168.
[32] T. Engel, K. Fehre, A. Rappelsberger, K.-P. Adlassnig, Qualitätssicherung in der Geburtshilfe durch leitlinienbasierte ntscheidungsunterstützung., in: Proceedings of eHealth2011, Austrian Computer Society, 2011, 313 - 318.
[33] W. Seeling, Computergestützte Therapieplanung von Immunsuppressiva bei Patienten mit Nierentransplantation, University of Technology, Vienna, 2011.
[34] N. Anastassova, Development and Evaluation of TempToxopert - Knowledge-based computer-aided decision support during prenatal screening for infection with Toxoplasma gondii, Medical University of Vienna, 2005.
[35] R.A. Jenders, R. Corman, B. Dasgupta, Making the standard more standard: a data and query model for knowledge representation in the Arden syntax, AMIA Annu Symp Proc. (2003) 323-330.
[36] GELLO - HL7Wiki. Available: http://wiki.hl7.org/index.php?title=GELLO (last accessed: January 31, 2012)
[37] K. Kawamoto, G. Del Fiol, D.F. Lobach, R.A. Jenders, Standards for Scalable Clinical Decision Support: Need, Current and Emerging Standards, Gaps, and Proposal for Progress, Open Med Inform J. 4 (2010) 235-244.
[38] S. Liu, Wei Ma, R. Moore, V. Ganesan, S. Nelson, RxNorm: prescription for electronic drug information exchange, IT Prof. 7 (2005) 17-23.

[39] Virtual Medical Record (vMR) - HL7Wiki. Available: http://wiki.hl7.org/index.php?title=Virtual_Medical_Record_(vMR) (last accessed: January 31, 2012)
[40] OWL2 Web Ontology Language Primer. Available: http://www.w3.org/TR/2009/REC-owl2-primer-20091027/ (last accessed: January 31, 2012)
[41] Product CTS - HL7Wiki. Available: http://wiki.hl7.org/index.php?title=Product_CTS (last accessed: January 31, 2012)
[42] R.N. Shiffman, G. Michel, A. Essaihi, E. Thornquist, Bridging the Guideline Implementation Gap: A Systematic, Document-Centered Approach to Guideline Implementation, J Am Med Inform Assoc. 11 (2004) 418-426.
[43] Decision Support Service (DSS). Available: http://www.hl7.org/v3ballot/html/infrastructure/dss/dss.html (last accessed: January 31, 2012)
[44] A.A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q.T. Zeng, D. Wang, u. a., GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines, J Biomed Inform. 37 (2004) 147-161.
[45] M. Peleg, A.A. Boxwala, E. Bernstam, S. Tu, R.A. Greenes, E.H. Shortliffe, Sharable Representation of Clinical Guidelines in GLIF: Relationship to the Arden Syntax, Journal of Biomedical Informatics. 34 (2001) 170-181.
[46] RIF Overview. Available: http://www.w3.org/TR/rif-overview/ (last accessed: January 31, 2012)
[47] A. Wright, D.F. Sittig, A Framework and Model for Evaluating Clinical Decision Support Architectures, J Biomed Inform. 41 (2008) 982-990.