Building Continents of Knowledge in Oceans of Data: The Future of Co-Created eHealth
A. Ugon et al. (Eds.)

doi:10.3233/978-1-61499-852-5-675

Generalizing the Arden Syntax to a Common Clinical Application Language

Stefan KRAUS [a,1]
[a] *Medical Informatics, University Erlangen-Nürnberg, Erlangen, Germany.*

Abstract. The Arden Syntax for Medical Logic Systems is a standard for encoding and sharing knowledge in the form of Medical Logic Modules (MLMs). Although the Arden Syntax has been designed to meet the requirements of data-driven clinical event monitoring, multiple studies suggest that its language constructs may be suitable for use outside the intended application area and even as a common clinical application language. Such a broader context, however, requires to reconsider some language features. The purpose of this paper is to outline the related modifications on the basis of a generalized Arden Syntax version. The implemented prototype provides multiple adjustments to the standard, such as an option to use programming language constructs without the frame-like MLM structure, a JSON compliant data type system, a means to use MLMs as user-defined functions, and native support of restful web services with integrated data mapping. This study does not aim to promote an actually new language, but a more generic version of the proven Arden Syntax standard. Such an easy-to-understand domain-specific language for common clinical applications might cover multiple additional medical subdomains and serve as a lingua franca for arbitrary clinical algorithms, therefore avoiding a patchwork of multiple all-purpose languages between, and even within, institutions.

Keywords. Domain-specific language, Arden Syntax, Medical Logic Modules

1. Introduction

The Arden Syntax for Medical Logic Systems is a Health Level 7 standard for encoding and sharing knowledge in the form of Medical Logic Modules (MLMs) [1]. In contrast to all-purpose programming languages such as Java or Python, the Arden Syntax is specifically designed to meet the requirements of data-driven clinical event monitoring [2]. Such special-purpose languages are commonly referred to as domain-specific languages (DSLs). An MLM typically makes a single clinical decision; it corresponds to a production rule insofar as it evaluates a condition and possibly performs an action. Each MLM has a mandatory frame-like structure, consisting of categories and slots. The behavior of an MLM at runtime is determined by procedural programming language constructs within specific slots. Consequently, an MLM is both a program and a production rule, also known as the hybrid character of the Arden Syntax.

In 2006, University Hospital Erlangen (UHER) introduced a commercial patient data management system (PDMS); at the largest intensive care unit (ICU) with currently 35 beds, a commercial Arden Syntax environment has been integrated [3]. Ever since the PDMS is in routine use, clinicians repeatedly demanded various kinds of add-ons to process electronic medical record (EMR) data, such as specific views of microbiology

[1] Corresponding Author, stefan.d.kraus@fau.de

results, billing support functions, or the calculation of multiple ICU-related clinical scores in the form of a graphically displayed time series. Although most of these add-ons were clearly outside the original scope of the Arden Syntax, its language constructs nevertheless proved mostly suitable for such purposes. The rich set of operators and the timestamped data type system, both tailored to EMR data, resulted in concise and easy-to-understand code. Therefore, we increasingly used the Arden Syntax in multiple other subdomains, such as information retrieval at the bedside, or patient recruiting. However, some features of the Arden Syntax that are suitable for its original purpose need to be revised with regard to additional application areas. The modifications resulted in a generalized version of the Arden Syntax that may be used as a DSL for common clinical applications. The purpose of this paper is to introduce this language, to outline those features that differ from the standard, and to discuss the necessity for such a DSL.

2. Methods

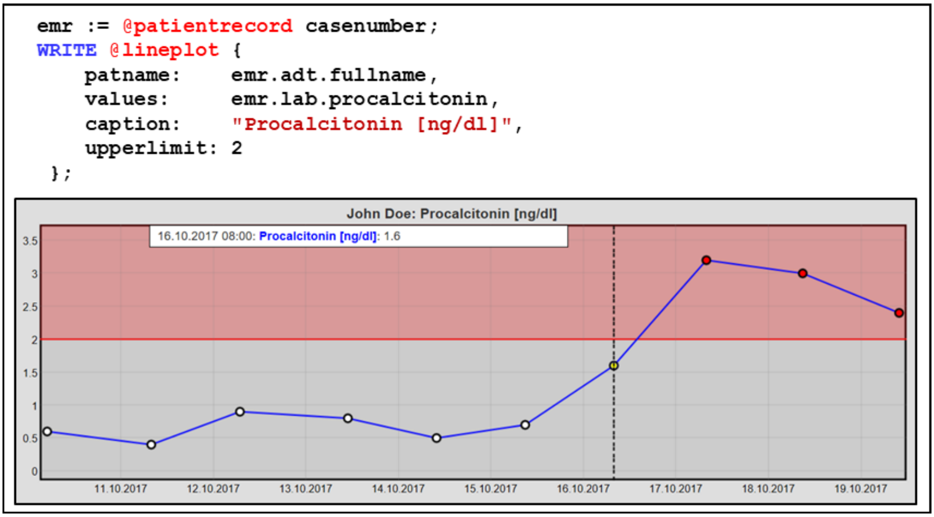
A prototype of a domain-specific language for common clinical applications has been created in the form of a generalized Arden Syntax version. The language was named ***P**rogramming **L**anguage, **A**rden-**IN**spired* (PLAIN) to point out its origin. Syntax and semantics of statements and operators have been defined to largely comply with version 2.8 of the standard [4]; several language features, however, have been altered, and some new conceptual approaches have been introduced, such as a means to use an MLM in arbitrary expressions as a user-defined function (UDF), identified by an MLM name prefixed with "@" (see example in Figure 1). The execution environment has been implemented in the form of a compiler and a virtual machine. Most parts have been written in the Haxe programming language [5], which supports the translation of the prototype into multiple all-purpose target languages.

The grammar provided by the Arden Syntax specification has been rewritten from scratch to remove some ambiguities. The definition of the frame-like MLM structure has been relocated from the grammar to a frame detector component that is capable of defining arbitrary alternative structures. The data type system has been harmonized with the JavaScript Object Notation (JSON) format, whereby the handling of time stamps has been retained. In contrast to the Arden Syntax, JSON supports nested lists, so two additional operators have been introduced to support their handling. To provide access to EMRs from network resources, native support of restful web services has been integrated, complemented with a data mapper that automatically transforms the query results into congruent PLAIN data types. To support a convenient mapping of entire EMRs, a minimalist PLAIN Data Markup Language (PDML) has been specified, which largely complies with the approach described in [6].

3. Results

The prototype of PLAIN is an application program, including a lightweight development environment designed for users with little programming knowledge. The most obvious difference from the Arden Syntax standard is that the frame-like MLM structure can be omitted, enabling “free-style” code that supports the benefits of well-tried statements and operators without the complexity of complete MLMs. While the Arden Syntax standard is limited to the frame type *data-driven*, PLAIN supports alternative types, such as *user-*

driven, or *sub-module* for code reuse. Additional types can be defined using the frame detector, where appropriate. In case MLMs invoke each other, MLMs with and without frame structure can be mixed.



```
emr := @patientrecord casenumber;
WRITE @lineplot {
    patname:     emr.adt.fullname,
    values:      emr.lab.procalcitonin,
    caption:     "Procalcitonin [ng/dl]",
    upperlimit: 2
  };
```

Figure 1. Basic PLAIN example MLM comprising two statements. The first statement reads an entire patient record and maps its contents to the variable *emr*. The variable *casenumber* is automatically set in any MLM that has a single patient context. The second statement graphically displays the patient's procalcitonin values.

The basic example MLM in Figure 1 shows multiple features that distinguish PLAIN from the Arden Syntax standard. It contains free-style code without frame structure. The first statement reads an entire EMR, using the UDF *patientrecord*, which internally applies a REST statement to retrieve the EMR for the specific case number from a network resource. The variable *casenumber* is automatically set in any MLM that has a single patient context. The retrieved EMR is mapped onto the variable *emr*, whereby its internal structure is fully preserved. The data mapper natively supports PDML and JSON encoded data structures. The second statement graphically displays the patient's procalcitonin values, which the EMR contains in the form of a time series. It utilizes the UDF *lineplot*, which takes a JSON encoded object as the argument. An object instantiated by means of this notation is of the generic type *structure*. In contrast to the static Arden Syntax object type, the attributes of such an object can be extended at runtime. This enables MLMs for generic and fully reusable preprocessing of patient data. After about one year of using PLAIN for research and teaching at the local chair of medical informatics, it replaced the Arden Syntax in clinical routine at UHER.

4. Discussion

This study introduces a generalized version of the Arden Syntax, motivated by the repeated observation that this standard might be beneficial in many application areas beyond its originally intended scope. The literature provides multiple examples of such an "off-label" use. Although Arden Syntax MLMs have been specifically designed for modularly independent data-driven event monitoring functions, they have been used for clinical protocols and care plans [7, 8], user-driven decision support functions [3, 9, 10], multi-patient clinical dashboards [11], patient recruiting [12, 13], discharge letters [14], and telemedical training of patients [15]. Such attempts may at least indicate a demand for a generalized version. The personal view of an individual user regarding this matter will mostly depend on his programming skills, the intended application area, and the computational environment at the specific institution.

Only a minority of those working in medical informatics are trained information scientists. Many colleagues originating from other disciplines frankly admit that they are not willing or able to learn an all-purpose programming language, but nevertheless show considerable interest in a simple and easy-to-learn programming language tailored to the medical domain. Algorithms used in clinical routine are not necessarily of high complexity. An easy-to-understand medical informatics DSL, such as a generalized version of the Arden Syntax, may be used as a lingua franca for such programming tasks, therefore avoiding a patchwork of multiple all-purpose languages between, and even within, institutions. The Arden Syntax itself may be seen as a lingua franca with respect to clinical decision support functions, especially in the context of data-driven event monitoring. A more generic Arden Syntax version may cover multiple additional medical subdomains, such as billing support, information retrieval, clinical dashboards, or patient recruiting. Moreover, it may decrease the dependence on healthcare information system vendors, as an integrated medical informatics DSL could provide the customer with a simple yet generic tool for defining versatile customized add-ons.

Whenever a new technology is introduced, one should be aware of the ubiquitous danger of creating "yet another standard" that hardly anyone uses in the end. However, PLAIN is not a new technology, but closely related with the proven Arden Syntax standard, which has been constantly improved during its evolution and is in routine use for more than two decades [16]. Furthermore, the number of publications in recent years reflects a continued interest in this standard, and the prototype described in this study is a result of an actual clinical demand. Representatives from several universities in German-speaking countries expressed their interest in a simple but generic medical informatics DSL, and the further development of PLAIN will be based on continued cooperation with periodic feedback, under the supervision of the working group for knowledge-based systems of the German Association for Medical Informatics.

The overall extent of the adjustments to generalize the Arden Syntax is limited. An important modification refers to the frame-like MLM structure. In Arden Syntax, each MLM behaves like a production rule, which is reflected in the characteristics of those slots that contain programming language constructs. For data-driven monitoring functions, a production rule behavior is beneficial. For several other use cases, this behavior is needless, like in case of user-driven information retrieval. In case MLMs process multiple EMRs, for example in patient recruitment or clinical dashboards, this behavior can even be obstructive. The resulting separation of the frame-like structure from the other language constructs considerably simplified the grammar. Such a separation could likewise simplify the maintenance of the Arden Syntax. It is at least conceivable that some characteristics of PLAIN could have an impact on the further evolution of the Arden Syntax standard.

Some of the differences between PLAIN and the Arden Syntax may require additional user skills. For example, PLAIN supports nested lists, while the designers of the Arden Syntax decided not to support them because they are difficult to handle and not necessarily required for data-driven clinical event monitoring. However, in more generic algorithms that process complex data structures such as lists of entire EMRs, the list handling of the Arden Syntax (many operators implicitly process lists by element) would in some cases unavoidably result in nested lists. Thus, a more generic language must be able to handle such situations that will rarely occur within the original application range of MLMs. There may be other solutions to the issue of potentially nested lists and some other language details that may require reconsideration, and PLAIN is also intended as a testing area for future research in order to find an optimum in a spectrum of potential solutions, in particular to find a balance between computational performance and ease-of-use. Currently, there are multiple ongoing investigations in this field that concentrate on specific details, such as the handling of timestamps in medical calculations, the introduction of a specific time range data type, the creation of clinical dashboards, multi-patient reporting functions, and patient phenotyping in clinical trials.

References

[1] Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res. 1994;27:291–324.

[2] Hripcsak G, Clayton PD, Jenders RA, Cimino JJ, Johnson SB. Design of a clinical event monitor. Comput Biomed Res. 1996;29:194–221.

[3] Kraus S, Castellanos I, Toddenroth D, Prokosch H-U, Burkle T. Integrating Arden-Syntax-based clinical decision support with extended presentation formats into a commercial patient data management system. J Clin Monit Comput. 2014;28:465–73. doi:10.1007/s10877-013-9430-0.

[4] Health Level Seven International: The Arden Syntax for Medical Logic systems, Version 2.8, 2012.

[5] Favre L, Pereria C, Martinez L. Modernizing software in science and engineering: From C/C++ applications to mobile platforms. In: Papadrakakis M, Papadopoulos V, Stefanou G, Plevris V, editors; Crete Island, Greece, 5–10 June 2016; 2016. doi:10.7712/100016.2402.4906.

[6] Kraus S, Enders M, Prokosch H-U, Castellanos I, Lenz R, Sedlmayr M. Accessing complex patient data from Arden Syntax Medical Logic Modules. Artif Intell Med 2015. doi:10.1016/j.artmed.2015.09.003.

[7] Sherman EH, Hripcsak G, Starren J, Jenders RA, Clayton P. Using intermediate states to improve the ability of the Arden Syntax to implement care plans and reuse knowledge. Proc Annu Symp Comput Appl Med Care. 1995:238–42.

[8] Seitinger A, Rappelsberger A, Leitich H, Binder M, Adlassnig K-P. Executable medical guidelines with Arden Syntax-Applications in dermatology and obstetrics. Artif Intell Med 2016. doi:10.1016/j.artmed.2016.08.003.

[9] Karlsson D, Ekdahl C, Wigertz O, Shahsavar N, Gill H, Forsum U. Extended telemedical consultation using Arden Syntax based decision support, hypertext and WWW technique. Methods Inf Med. 1997;36:108–14.

[10] Castellanos I, Kraus S, Toddenroth D, Prokosch H-U, Burkle T. Using Arden Syntax Medical Logic Modules to reduce overutilization of laboratory tests for detection of bacterial infections-Success or failure? Artif Intell Med 2015. doi:10.1016/j.artmed.2015.09.005.

[11] Kraus S, Drescher C, Sedlmayr M, Castellanos I, Prokosch H-U, Toddenroth D. Using Arden Syntax for the creation of a multi-patient surveillance dashboard. Artif Intell Med 2015. doi:10.1016/j.artmed.2015.09.009.

[12] Ohno-Machado L, Wang SJ, Mar P, Boxwala AA. Decision support for clinical trial eligibility determination in breast cancer. Proc AMIA Symp. 1999:340–4.

[13] 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;2014:1028–36.

[14] Kraus S, Castellanos I, Albermann M, Schuettler C, Prokosch H-U, Staudigel M, Toddenroth D. Using Arden Syntax for the Generation of Intelligent Intensive Care Discharge Letters. Stud Health Technol Inform. 2016;228:471–5.

[15] Song B, Wolf K-H, Gietzelt M, Al Scharaa O, Tegtbur U, Haux R, Marschollek M. Decision support for teletraining of COPD patients. Methods Inf Med. 2010;49:96–102. doi:10.3414/ME09-02-0005.

[16] Jenders RA, Adlassnig K-P, Fehre K, Haug P. Evolution of the Arden Syntax: Key Technical Issues from the Standards Development Organization Perspective. Artif Intell Med 2016. doi:10.1016/j.artmed.2016.08.001.