Peter Seifter^{1,2*}, Stefan Sabutsch^{2,3}, Jeroen S. De Bruin^{1,4}

¹Institute of eHealth, Department of Applied Computer Science, FH Joanneum University of Applied Computer Sciences GmbH, Graz, Austria

² HL7 Austria, Austria

³ ELGA GmbH, Austria

⁴ Section for Artificial Intelligence and Decision Support, Center for Medical Statistics, Informatics, and Intelligent Systems, Medical University of Vienna, Vienna, Austria

Abstract

Background: Recently, the Austrian Patient Summary (APS) and Austrian Microbiology Report (AMR) document definitions were added to the Austrian national electronic health record repository. How to employ these standardized, structured patient data in clinical decision support (CDS) systems remains an interesting research subject.

Objectives: To create a proof of concept for the emergency use of APS and AMR documents in a variety of CDS platforms and standards in an intensive care setting, in order to generate new insights by linking individual documents to different CDS approaches.

Methods: APS and AMR definitions were stored using MongoDB, a document-oriented database system. For implementation of the clinical use case CDS, we selected the Drools business rule management system, as well as the ArdenSuite software for implementing the HL7 Arden Syntax for Medical Logic Systems standard.

Results: Due to its manifold features, implementation of the CDS use case in Drools could be done quite efficiently

Correspondence to:

Dr. Peter Seifter

Institute of eHealth, Department of Applied Computer Science, FH Joanneum University of Applied Computer Sciences GmbH, Graz, Austria.

E-mail: peter.seifter@fh-joanneum.at

1 Introduction

As the volumes of medical data generated in healthcare become greater, and the heterogeneity of these data continues to increase as well, the need for support in evaluation and interpretation of these data increases as well. As a result, the demand for clinical decision support (CDS) systems has

without the need for additional coding of technical or algorithmic code, which results in smaller rules. In contrast, the use case implementations in Arden Syntax using the ArdenSuite required additional technical coding, distracting from the medical knowledge implementation. The Arden Syntax as a knowledge definition standard is, however, better understandable due to its resemblance to natural language compared to the Java-like definition language used in Drools.

Conclusion: With the nation-wide availability of structured documents, the foundation for the implementation of clinical decision support systems has been laid. Commonly used open-source platforms offer extensive possibilities for the implementation of CDS systems. This evaluation of modern business rule management systems will advance the implementation of powerful clinical decision support solutions valuable to all stakeholders.

Keywords

Clinical document architecture; Patient summary; Microbiology laboratory report; Decision support systems; Clinical; Decision support techniques

EJBI 2018; 14(3):21-31 Received: March 19, 2018 Accepted: May 19, 2018 Published: July 06, 2018

steadily increased. Over the years, various CDS systems for a large variety of medical specialties and purposes have been created, with varying success, e.g., in infection control [1, 2].

One of the prerequisites for creating successful, interoperable CDS systems is the availability of structured, standardized data sources. Structure in data improves its usability in digital processing, including CDS system, while semantic standardization through the use of code standards and value sets allow interpretation across healthcare institutes, and even across borders. In our approach, we restricted data sources to documents structured in the Clinical Document Architecture (CDA), an international document markup standard that specifies the structure and semantics of clinical documents for the purpose of exchange between healthcare providers and patients [3]. CDA documents are easy to use because of their in machinereadable format and well-known access methods. Furthermore, due to their XML-based encoding, extraction of individual data elements is quite easy as well, and supported by standardized methods on many development platforms, including Java.

Whereas interoperability standards like CDA express structured aspects of a patient's health, (e.g., a test result, or an overview of a patient's allergies), clinical decision support can be used to create connections between these structured data elements in order to generate new information. In this process, new, higher-level insights in a patient's health or comprehensive views on a patient's health in a specific, expert context, e.g., infection control, are provided through combination and interpretation of data element from individual documents and medical knowledge. To this end, basic elements of a CDA document have to be transformed - depending on the context - for further examination and linkage with other data sources. Furthermore, based on this kind of representation of the source data, a formal knowledge representation system is required for the generation of new knowledge.

In this study, we explore the use of CDA documents for clinical decision support. To this end, we combined two types of CDA-based document standards, the Austrian Patient Summary and the Austrian microbiology lab report. The Austrian Patient Summary contains essential healthcare information intended for unscheduled (e.g. emergency) use, whereas the Austrian microbiology laboratory report provides relevant information for the observation and therapy of bacterial or other microbiological infections. Given an infection control use case in an intensive care setting, we demonstrate how decision support can be applied to combine aforementioned data sources. To enable comparison between different knowledge representation standards in decision support, we implemented the use case in both Arden Syntax [4], a knowledge-based clinical decision support system standard, and in Drools [5], a general-purpose business rule management framework.

2 Methods

2.1 Clinical Document Interoperability Standards

Central to this study is the combination of heterogeneous medical information, which is spread over different clinical documents. CDA is the key to the harmonization of health data structure and CDA Release 2 was stipulated as the relevant document standard. In a nationwide specification process for these CDA documents the main stakeholders of the Austrian health system have developed so called "CDA Implementation Guides" for classes of documents like the "Physician's Discharge Summary", the "Nursing Discharge Summary", the "Laboratory Report" and the "Diagnostic Imaging Report" on a consensual basis.

In this section we provide a short overview of other standard documents, which are used in this study (a comprehensive discussion falls outside of the study scope):

- The Austrian Patient Summary document based on the International Patient Summary document and
- The Austrian microbiology lab report.

International and Austrian Patient Summary: According to HL7 IPS [6], the International Patient Summary (IPS) document is "an electronic health record extract containing essential healthcare information intended for use in the unscheduled or unplanned, cross-border care scenario". In this context, the scope of "essential healthcare information" is defined by the required elements of the IPS dataset, which is a specialty-agnostic, condition-independent minimal and non-exhaustive patient summary dataset, readily usable by clinicians.

The current IPS implementation comprises a set of templates and profiles based on HL7 Clinical Document Architecture (CDA) Release 2 [3] and Fast Healthcare Interoperability Resources (FHIR) [7] resource profiles, with value sets to support standardized coding of data elements. As the use of IPS documents is cross-jurisdictional in nature (both on an international and national level), this implies a need for common templates, and supported value sets based on international (multi-lingual) vocabularies [8].

In 2005 the foundation was laid for the establishment of an Austrian-wide electronic health record system (German acronym: ELGA). Since then, a variety of ELGA-supported structured, standardized document templates [9] were conceived and implemented in the majority of hospitals and at general practitioners, including document templates and resource definitions for electronic prescription, discharge summaries, laboratory report and radiology reports.

The Austrian Patient Summary (APS) [10] document is the latest document structure to have been developed. A patient summary working group was formed which met between 2016 and 2017 and, under the auspices of ELGA management, harmonized the APS content requirements and the terminologies to be used. The resulting APS document definition contains provisions for general demographic patient data (e.g. name, date of birth, gender), a summary of the insights and contents from the medical records of the patient (e.g. current medical problems, allergies and the degree of intolerance, major surgical interventions, medical implants, inoculations) as well as the current medication. A sample of an APS document is shown in Figure 1.

In accordance with the IPS document definition, APS documents are structured XML documents that comply with the HL7 CDA R2 standard. The CDA Implementation Guide for the APS was created using ART-DÉCOR [11], an open-source tool that supports the creation and maintenance of HL7 templates, value sets, scenarios and data sets. The technical specification is based on the IPS Implementation Guide [6] and was subsequently published via a wiki.

The Austrian Microbiology Lab Report: The CDA implementation guide for the Austrian microbiology laboratory report (AMR) is an extension of the existing CDA laboratory report, which is based on the Integrating the Healthcare Enterprise (IHE) Sharing Laboratory Reports (XD-LAB) Integration Profile [12]. The AMR enables the caregiver to obtain relevant information for the observation and therapy of bacterial or other microbiological infections and to import the structured and coded data into their electronic health record (EHR). At the time of writing, the implementation guide is subject to the normative ballot process of HL7 Austria.

The AMR implementation guide includes sections on general report information, information on the collected test sample, and microbiological laboratory results. The general report information section provides administrative information clinical context on the reported result, e.g., the date of order entry, a patient's suspected diagnosis, requested examinations, and comments on report findings. The section on the collected test sample contains information on the test specimen, e.g., its time of collection, material type, the procedure with which it was extracted, as well as comments on the specimen quality. Finally, the results section contains all relevant result information, e.g., microscopic and macroscopic information, detected pathogens and their antibiotic resistance patterns, infection serology, etc. A document sample of the AMR is shown in Figure 2.

For the standardization of entry values, various coding mechanisms have been used, including SNOMED CT [13], LOINC [14], HL7 value sets (e.g., observation-interpretation, specimen type, ...) and ELGA value sets used to complement aforementioned code systems.



| reated 30. Juli 2015 | at 13:01 version: 1 | | | | | | | |
|--|--|---|---|---|--|--|--|--|
| Expand the table | of contents 💌 Exp | and all contents | | | 2 | | | |
| Patient: | Dr. Maria Johanna Musterfrau, BSc Sex: female born: 24. Dezember 1961 SSN: 1111241261 | | | | | | | |
| Client: Host | pital Unterstadt | | | | | | | |
| Created by: | Amadeus Hospital - | Laboratory | To: C | Ordination Dr. Doctor | | | | |
| usuales the lechnic | a possibilities using a | maximum or the un | owed options. | | | | | |
| eason for refer (Text of the referrir equested inves Anaerobic culture | ral Ig physician) clarificati stigations | on of a bacterial inf | ection (anaerobes?) | | | | | |
| Anaerobic culture Anaerobic culture Anaerobic culture Anaerobic not culture | ral g physician) clarificati stigations tion Sampling | Investigated material | ection (anaerobes?) Sampling by | sample input | Remark laboratory | | | |
| teason for refer (Text of the referrin requested invest Anaerobic culture ample informat Material ID BL-121201-02 | ral g physician) clarificati stigations Sampling 01.12.2012 06:34 | Investigated material Blood, fossa cubitalis I. | ection (anaerobes?) Sampling by Dr. Humpel | sample input 01.12.2012 08:15 | Remark laboratory slightly hemolytic | | | |
| eason for refer (Text of the referrir equested invest Anaerobic culture ample informat Material ID BL-121201-02 lacroscopy This section descri The result is docur licroscopy Characteristics | ral g physician) clarificati stigations tion 01.12.2012 06:34 bes the nature of the s nented as free text. | Investigated material Blood, fossa cubitalis I. Submitted material, Result | ection (anaerobes?) Sampling by Dr. Humpel which was detected t | sample input 01.12.2012 08:15 with the eye (without the Unit | Remark laboratory slightly hemolytic ald of a microscope) | | | |

2.2 Methods and Tools for Knowledge Definition based user interface that allows authoring and management and Inference

system. Knowledge-based systems are production rule systems, for which a collection of rules and restrictions are defined. These Rete-OO algorithm [15, 16] and backward chaining (goalrules are then evaluated with actual patient data (facts) using an driven analysis). Using these two tools and the Drools core, inference mechanism.

After evaluation of different tools and standards, we chose to implement our CDS use case with one of the more popular general-purpose, open-source business rule management systems (BRMS) called Drools [5].

Apart from using a general-purpose tool, we also implemented our use case in a standard developed specifically for knowledgebased clinical decision support systems as well: the HL7 Arden Syntax for Medical Logic Systems [4].

The Business Rule Management Platform Drools: Drools is an open-source BRMS platform implemented in Java. Without going too much into the intricacies of Drools, the platform can be described as a collection of tools that permits the decoupling of data and logic, and allows reasoning over data within various business processes, including medical reasoning. As such, Drools requires that a data model is created on which rules are applied. In turn, declarative rules are defined that express constraints on executed, after which results are returned to the Arden Syntax the data model.

For this study, two Drools components are of especial importance. First, the Drools Workbench, which is the web- support technology platform for the implementation and

of business rules and the underlying data model. Second, the Drools Expert module, which is the declarative, rule based, For our CDS use case we created a knowledge-based CDS coding and execution environment, which implements both forward chaining (data-driven analysis, based on the Rete / we defined the data model for both the APS and the AMR and implemented the clinical use case.

> ArdenSuite CDS platform: Arden Syntax is an HL7 standard for the computerized representation and processing of medical knowledge, e.g., treatment rules, diagnostic decision trees, and risk scores. An Arden Syntax knowledge base commonly consists of multiple modules, called medical logic modules (MLMs) [17, 18]; these MLMs partition the knowledge base in highly-cohesive knowledge artifacts, where each MLM should contain logic pertaining to a single medical decision.

> In contrast to Drools, Arden Syntax makes no assumptions about an underlying data model. Instead, in Arden Syntax external data resources can be accessed using read or write statements, or within curly braces, where larger data query and retrieval operations can be defined. Execution of these operations is forwarded to the host system, where it is MLMs.

For this study, we used the ArdenSuite clinical decision

an ArdenSuite server which is used for the storage, management, informed decision. Furthermore, in the second use case, and execution of MLMs. Furthermore, it contains an ArdenSuite a second warning is generated if an antibiotic is selected to integrated development and test environment (IDE), which which the pathogen has increased resistance (encoded in the serves as an authoring and test tool for Arden Syntax MLMs. To data as either "intermediary" or "resistant"). access external data sources, the ArdenSuite comes equipped with a standard connector for Java Database Connectivity (JDBC)- 3 compatible databases.

2.3 Clinical Use Case

unresponsive patient required emergency heart surgery and was primary reasons were both the availability of a Java driver transferred to the intensive care unit (ICU) afterwards. After and a JDBC connector, its ability to store data in flexible three days, the patient - still unresponsive - develops clinical document that can change over time ("schema-free"), and its infection symptoms, i.e., fever, increased need for vasopressin consistent performance in the management of complex CDA (noradrenaline). Increased infection parameters (elevated leucocyte counts and C-Reactive protein). To confirm suspicions of sepsis and determine its source, blood samples are taken and sent to the department of clinical microbiology. Simultaneously, the surgeon starts a broad-spectrum antimicrobial therapy (either amoxicillin or meropenem). Two days later, the microbiology test results confirm the presence of Methicillin-Resistant Staphylococcus Aaureus (MRSA), upon which the treatment is serialized them as JSON objects to the MongoDB database. changed to either vancomycin or daptomycin.

data use for both the IPS/APS and the AMR. In this case, essential type. Data access to the MongoDB database depended on the healthcare information is provided for unscheduled (emergency) CDS implementation method (Drools or Arden Syntax). A use. Despite the patient being in an unresponsive state, his/her graphical depiction of the process described above in Figure allergies to various antibiotics, if any, are available in the IPS/APS. 3. This provides invaluable information for both the initial broadspectrum antibiotics treatment and the later treatment for MRSA: The AMR also provides essential, standardized information on antibiotic susceptibility of the MRSA pathogen which can be compared with available allergy information.

In this study, we implemented alert rules for both decision making scenarios (with respect to antimicrobial therapy) in a knowledge base:

- 1. The selection of broad-spectrum antimicrobial therapy, motivated and informed by allergies reported in the IPS/ APS and the antimicrobial agent selected by the attending physician, and
- 2. The selection of antimicrobial therapy for MRSA, based on 3.1 Drools Implementation the antimicrobial resistance patterns of the MRSA pathogen reported in the Austrian microbiology report, the allergies reported in the IPS/APS and the antimicrobial agent selected by the attending physician.

For both scenarios, alerts are implemented that warn the physician in case an allergy to an antimicrobial agent is present; rule engine (Figures 4a and 4b) show the message definitions the alert also presents the severity of the intolerance (as shown for the APS and the AMR.

execution of our clinical use case [19]. The ArdenSuite comprises earlier in Figure 1) to allow the physician to make a fully

Results

For document storage, we used the MongoDB Community Edition [20], a free, open-source, document-oriented alternative for the management of HL7 CDA documents. Consider the following (simplified) clinical use case: An Although there are various benefits to using MongoDB, our documents [21].

> Data import from both CDA-based document and subsequent translation to Binary JSON (BSON) format was done using the open-source Model-Driven Health Tools (MDHT) library [22]. Using this library, we created classes for relevant information elements for both APS and AMR, instantiated them for each document in our test data, and

Separate collections (collections are analogues to tables The aforementioned scenario falls in the scope of intended in relational databases) were created for each document

> For the implementation of the use case, we used a simplified approach as a proof of concept. In the use case scenarios, medication names for patient allergies and pathogen antimicrobial resistance patterns are matched against a single antibiotic name proposed by the physician. A more advanced approach would take into account transformation of names to and from an ontology or thesaurus of medication families, e.g., as is defined in the Anatomical Therapeutic Chemical (ATC) Classification System, whereby we take into account allergies or resistances to a family of medications or active ingredients in a medication. This approach was deliberately omitted in this paper, and left for publication of the clinical use case implementation at a later stage.

For the implementation of our clinical use case in Drools, we first had to implement persistent objects for the underlying data model based on the APS and AMR document definitions. Based on these object definitions, we automatically generated facts (in the form of messages) that were inserted in the Drools



```
public static class APSMessage {
                                         public static class AMRMessage {
    private String Substance;
                                             private String Pathogen;
   private String Type;
                                             private String Substance;
   private String Reaction;
   private String Intensity;
                                             private String Susceptib;
   private String Severity;
                                             private AustrianMicrobiologyReport amr;
    private AustrianPatientSummary aps;
                                             // Readers and writers...
    // Readers and writers...
                                             // ...
    // ...
```

Figure 4(a): Austrian Patient Summary fact definition in Java / Drools (4b). Austrian Microbiology Report fact definition in Java / Drools.

To generate facts for Drools, we wrote supporter functions that would unravel list and other containers in Java and thus create a number of facts by Cartesian product. An example: If a new microbiology report was detected, in which antimicrobial resistance patterns for m pathogens were tested with n different antimicrobial agents, then m*n facts were inserted; each fact contains a pathogen name, an antimicrobial substance, and a resistance indicator.

Because of the supporter functions, the knowledge base itself could be implemented in a straightforward fashion. Figure

5 shows the implementation of the first use case decision scenario, in a rule called "Find allergy". This rule is only fired if an antibiotic was proposed by the physician and if this antibiotic matches an allergy recorded in the Substance field of the APS message. If so, an *AllergyFound* message is created and passed on.

In this rule, the presence of a substance allergy (the member variable Substance of the message *\$apsm*) is checked against *\$antib*, the proposed antibiotic by the physician.

Similarly, a rule for the second decision scenario (named "Find resistance pattern") was defined, which is shown in Figure 6. Note that only the alert for increased resistance is implemented here; as Drools checks facts against all rules in the knowledge base, an explicit call to the first rule is not necessary.

In this rule *\$antib*, the proposed antibiotic by the physician, is checked against the resistance pattern of a pathogen against the Substance recorded in the AMRMessage *\$amrm*.

3.2 Arden Syntax Implementation

The resulting Arden Syntax knowledge base comprises four MLMs. Two MLMs provide supporting functions for both document types; for each document type, we implemented Arden Syntax object definitions, database query definitions and a function that transforms coded database query results into their decoded counterparts and returns those in the defined object. The remaining two MLMs implement the decision logic for both decision making scenarios in the use case using aforementioned

support functions. Database access was provided by configuring the ArdenSuite DBConnector for our MongoDB server.

Analog to our Drools implementation, we created objects that model APS and AMR information necessary to our decision making processes. Figures 7a and 7b show (relevant parts of) the Arden Syntax object definitions for respectively the APS and AMR. Using these objects, the two decision making scenarios were implemented. The logic in the MLM for the first scenario is shown in Figure 8. In this MLM, we have to explicitly check for identifications, as we don't have an underlying persistent data model and instead need to use queries. Furthermore, we have to perform various type checks in order to correctly handle the data. As a result, a (list of) allergy objects is returned if allergies were found, otherwise the MLM returns NULL (return code not shown). In this module, the presence of allergies is checked against the parameter antib, again representing the antibiotic agent proposed by the physician.

Figure 5: Drools rule "Find allergy" for the first use case decision scenario.

Figure 6: Drools rule "Find resistant pathogen" for the second use case decision scenario.

| AType, Reaction, Intensity, Severity | <pre>// reaction type (allergy, intolerance) // (List) of reactions // (List) of reaction intensities // seriousness (e.g., life-threatening)</pre> | 1: | Amount, Substance, Susceptib | | amount that was found (List) of substance test (List) of susceptibilites (S, I, R) |
|---|---|---------|------------------------------------|------------|--|
| iauro (| 7a), Austrian Dationt | " Su | mmar | • • • | object definition in |

Finally, in Figure 9, (part of) the MLM for the second scenario is shown. This MLM calls the previous MLM and additionally verifies that any pathogen found in the patient is not (intermediary) resistant to the antimicrobial agent proposed by the physician. Similar to the MLM in Figure 8, we again have to perform some type handling. As a result, a (list of) allergy objects and a list of resistance objects is returned if allergies or resistances were found, otherwise the MLM returns NULL (return code not shown).

In addition to calling the MLM previously discussed, this MLM also verifies that none of the found pathogens in the available microbiology reports are (intermediary) resistant to antib, the antibiotic agent proposed by the physician.

4 Discussion

In this paper we discussed the implementation of a clinical infection-control use case that employs patient data from two new standardized medical documents in Austria: The Austrian Patient Summary (based on the International Patient Summary) and the Austrian Microbiology Report. Data from these documents were extracted, and the clinical use case was implemented using two different knowledge definition standards: the general-purpose platform Drools, and HL7 Arden Syntax for Medical Logic Systems.

Both patient data document types are structured using CDA R2, but some data entries are optional. Furthermore, both document definitions are very recent and still subjective to change. Because of these sources of potential variability in document structure, we chose to store these documents in MongoDB, as there is no rigid schema definition, and because it allows for high loads due to its horizontal scalability (which is useful for data-intensive epidemiological applications). However, these benefits required that some traditional properties of classical data base management systems are changed or omitted (e.g., a lack of table joins). As of yet, little research has been done on the use of MongoDB in EHRs or for the storage of structured medical documents, but initial results are encouraging, showing good and consistent performance in the management of complex CDA documents, even in a non-optimized implementation [21].

```
// apsResults.Allergies can be an object or list of objects.
apsResults := CALL cdaApsMlm WITH lastName, firstName, DoB;
// No Allergies recorded
IF apsResults IS NOT PRESENT THEN
    allergyRetVal := NULL;
ELSEIF apsResults. Allergies IS NOT PRESENT THEN
    allergyRetVal := NULL;
// Only one allergy was found
ELSEIF apsResults. Allergies IS NOT LIST THEN
    IF apsResults.Allergies.Substance IS EQUAL antib THEN
        allergyRetVal := apsResults.Allergies;
    ENDIF;
// A list of allergies was returned
ELSE
    allergyRetVal := ();
    FOR allergy IN apsResults.Allergies DO
        IF allergy.Substance IS EQUAL antib THEN
            allergyRetVal := allergyRetVal , allergy;
        ENDIF;
    ENDDO:
ENDIF;
     Figure 8: Medical logic module for the first decision scenario.
```



Our choice for Drools as one of the platforms to implement our clinical use case was motivated by a short online study that we performed on popular open-source decision support tools. In the results of our online search, Drools appeared more often than other tools and platforms that we came across. Moreover, a search on PubMed showed that there are various publications on the successful implementation of CDS systems with Drools [23, 24, 25]. Our choice for Arden Syntax was a straightforward one and follows from the description and purpose of the standard: A clinical and scientific knowledge definition language that is used in a computer-executable format by clinical decision support systems [26].

When comparing our CDS implementations we have to distinguish between the implementation languages and their supporting frameworks. At first sight, the Drools implementation of our clinical use case seems shorter and easier, but this is for a large part because of the underlying Drools platform implementation. ArdenSuite and Drools offer similar functionalities i.e., both offer an IDE, web-based remote deployment, facilities for database access and workflow support in Business Process Modeling and Notation (BPMN). However, compared to the ArdenSuite, the Drools platform is richer in features and offers pre-implemented forward and backward chaining reasoning algorithms, which allow for shorter rule definitions without the need for additional technical or algorithmic implementations in the knowledge base.

A comparison of the knowledge definition languages themselves showed that the Drools rule definition language and its conventions remind strongly of the Perl language, and the Java programming language on which Drools is based. As such, it has a very technical appearance and is therefore harder to understand for those without a technical background. In this respect, the Arden Syntax has a clear advantage; the syntax was created for a broad, potentially non-technical public, thereby supporting operations not only tailored to use in the clinical realm, but also expressed in a syntax resembling narrative, natural language [26]. This makes understanding MLMs easier, which allows the MLMs themselves to serve as a communication device between knowledge engineer and clinician. By avoiding an intermediate representation, a potential source of translation errors is avoided.

There are various caveats and limitations to this study. Although the authors have a thorough understanding of CDA principles and the APS and AMR, and are proficient in the use of MongoDB, Drools and Arden Syntax, the various data model and knowledge base implementations provided in this study are not guaranteed to be optimal. The comparison provided in this study is a proof of concept that needs further evaluation. Given that only a small rule base has been implemented, the scalability of the used tools has to be verified, both for the data model implementation in MongoDB and certainly for the used CDS platforms and standards. Nonetheless, this study yielded useful experiences and knowledge on the implementation of CDS systems in combination with CDA-based document standards. Furthermore, this study generated hypotheses on combining implementations, possibly using the Arden Syntax a knowledge engineering tool and then translating it to Drools projects to take advantage of the strength of both the Arden Syntax and the powerful underlying Drools implementation. To the authors' knowledge, a proof of concept for such an undertaking exists, but was never followed up on [27].

5 Conclusion

The selection of methods and tools to generate new insights in a clinical environment based on medical data is a key factor for correct and efficient knowledge creation. The need to manage heterogeneous medical information requires the use of interoperability standards for clinical documents. This requirement is met due to the ELGA project and the standardization activities in Austria.

With the nation-wide availability of structured documents, the foundation for the implementation of clinical decision support systems has been laid. Furthermore, linking individual patient health documents enriches the quality and insight of structured patient data.

The implementation of the clinical use case in Drools could be done quite efficiently, and allows for easy entry into the world of medical information management, while the implementation in Arden Syntax produced software solutions which are easy to understand due to its resemblance to natural language.

As future work, we plan to extend and advance this evaluation of modern business rule management systems, in order to advance the implementation of more powerful clinical decision support solutions valuable to all stakeholders.

6 Acknowledgement

The authors are indebted to their colleagues from SDOs such as HL7 for open minded and kind cooperation and valuable support. Furthermore, the authors would like to thank Dr. Martin de Jonge of the Gesundheit Nord GmbH Klinikverbund Bremen for his invaluable assistance in specifying the clinical use case.

References

[1] De Bruin JS, Seeling W, Schuh C. Data use and effectiveness in electronic surveillance of healthcare associated infections in

the 21st century: a systematic review. J Am Med Inform Assoc. 2014; 21: 942-951.

- [2] Freeman R, Moore LSP, García Álvarez L, Charlett A, Holmes A. Advances in electronic surveillance for healthcare-associated infections in the 21st Century: a systematic review. J Hosp Infect. 2013; 84:106-119.
- [3] Health Level Seven International. HL7 Standards Product Brief - Clinical Document Architecture (CDA*) Release
 2. http://www.hl7.org/implement/standards/product_ brief.cfm?product_id=7
- [4] Health Level Seven International. HL7 Standards Product Brief - Arden Syntax v2.10 (Health Level Seven Arden Syntax for Medical Logic Systems, Version 2.10). http://www.hl7.org/implement/standards/product_brief. cfm?product_id=372
- [5] Red Hat, Inc. Drools-Business Rules Management System (JavaTM, Open Source). https://www.drools.org/
- [6] Health Level Seven International. International Patient Summary - HL7 IPS. http://internationalpatient-summary.net/mediawiki/index.php?title=IPS_ implementationguide_1
- [7] Health Level Seven International. Fast Healthcare Interoperability Resources (FHIR), Version 3 (Standard for Trial Use). https://www.hl7.org/fhir/
- [8] European Commission. International Patient Summary: Policy, deployment, competencies and standards. European Commission. https://ec.europa.eu/digitalsingle-market/en/news/international-patient-summarypolicy-deployment-competencies-and-standards
- [9] ELGA GmbH. Technical Basics. http://www.elga.gv.at/ index.php?id=2
- [10]HL7 Austria. ILF:Patient Summary Guide. https://wiki. hl7.at/index.php?title=ILF:Patient_Summary_Guide
- [11]ART-DECOR Expert Group. ART-DECOR*.https:// www.art-decor.org/mediawiki/index.php?title=Main_ Page
- [12] Integrating the Healthcare Enterprise (IHE). Laboratory Technical Framework. http://www.ihe.net/Technical_ Frameworks/#laboratory
- [13]Snomed International. SNOMED CT. https://www. snomed.org/snomed-ct
- [14] Regenstrief Institute. LOINC -The freely available standard for identifying health measurements, observations, and documents. https://loinc.org/
- [15]Forgy CL. Rete: A fast algorithm for the many pattern/ many object pattern match problem. Artif Intell. 1982; 19: 17-37.

- [16] Sottara D, Mello P, Proctor M. A Configurable Rete-OO Engine for Reasoning with Different Types of Imperfect Information. IEEE Trans Knowl Data Eng. 2010; 22: 1535-1548.
- [17] Pryor TA, Hripcsak G. The Arden syntax for medical logic modules. Int J Clin Monit Comput. 1993; 10: 215-224.
- [18] Hripcsak G. Writing Arden Syntax medical logic modules. Comput Biol Med. 1994; 24: 331-363.
- [19]Medexter Healthcare GmbH. ArdenSuite CDS Platform. https://www.medexter.com/products-and-services/ ardensuite
- [20] MongoDB Inc. MongoDB 3.6. MongoDB. https://www. mongodb.com/mongodb-3.6
- [21] Gomez A, Eijo J, Martínez von Scheidt M, Baum A, Luna D, Quirós F. MongoDB: An open source alternative for HL7-CDA clinical documents management. Proceedings of Open Source International Conference-CISL; 2013; Buenos Aires, Argentina.
- [22] Model Driven Health Tools (MDHT). SourceForge. https:// sourceforge.net/projects/oht-modeling/

- [23] Singh H, Yadav G, Mallaiah R, Joshi P, Joshi V, Kaur R, et al. iNICU-Integrated Neonatal Care Unit: Capturing Neonatal Journey in an Intelligent Data Way. J Med Syst. 2017; 41: 132.
- [24] Lardon J, Asfari H, Souvignet J, Trombert-Paviot B, Bousquet C. Improvement of Diagnosis Coding by Analysing EHR and Using Rule Engine: Application to the Chronic Kidney Disease. Stud Health Technol Inform. 2015; 210:120-124.
- [25] Ongenae F, De Backere F, Steurbaut K, Colpaert K, Kerckhove W, Decruyenaere J, et al. Towards computerizing intensive care sedation guidelines: design of a rule-based architecture for automated execution of clinical guidelines. BMC Med Inform Decis Mak. 2010; 10: 3.
- [26]Samwald M, Fehre K, de Bruin J, Adlassnig KP. The Arden Syntax standard for clinical decision support: Experiences and directions. J Biomed Inform. 2012; 45: 711-718.
- [27] Jung CY, Sward KA, Haug PJ. Executing medical logic modules expressed in ArdenML using Drools. J Am Med Inform Assoc. 2012; 19: 533-536.