Eine experimentelle Weiterentwicklung der Arden-Syntax

Dr. Stefan Kraus, Wissenschaftlicher Mitarbeiter am Lehrstuhl für Medizinische Informatik der Universität Erlangen-Nürnberg, stefan.d.kraus@fau.de

- Die Arden-Syntax ist ein Standard zur Wissensrepräsentation in Form von Medical Logic Modules (MLMs)
- MLMs sind für das Clinical Event Monitoring optimiert, lassen sich aber auch für andere Zwecke einsetzen
- Die Erfahrungen mit der Arden-Syntax am UK Erlangen führten zu einer experimentellen Variante namens PLAIN
- Die Weiterentwicklung soll im Rahmen der GMDS-Arbeitsgruppe Wissensbasierte Systeme konzertiert werden

Vorbemerkungen

Am Universitätsklinikum Erlangen (UKER) konnten über einen Zeitraum von etwa 10 Jahren umfangreiche Erfahrungen mit dem Einsatz der Arden-Syntax in der klinischen Routine gesammelt werden, sowohl auf Basis selbst entwickelter als auch kommerzieller Ausführungsplattformen. Aus der Bilanz dieser Erfahrungen entstand eine experimentelle Weiterentwicklung mit einigen Modifikationen und neu integrierten Leistungsmerkmalen. Im Oktober 2017 wurde die Arden-Syntax in der Routine am UKER durch die experimentelle Variante abgelöst. Der Artikel beschreibt die aktuellen Arbeiten in diesem Kontext.

Der Arden-Syntax-Standard

Die Arden Syntax for Medical Logic Systems ist ein HL7-Standard zur Repräsentation von medizinischem Wissen in Form sogenannter Medical Logic Modules (MLMs) [1]. Sie entstand in den späten 80ern [3] und wurde kontinuierlich weiterentwickelt [4], bis hin zur Unterstützung unscharfer Logic [5]. Die Arden-Syntax wurde in einer Reihe klinischer Informationssysteme integriert, teils in Form von Eigenentwicklungen im universitären Bereich, teils als integraler Bestandteil kommerzieller Produkte. Ein typisches MLM verhält sich wie eine WENN-DANN-Regel und beschränkt sich auf eine einzelne klinische Fragestellung. Ein simples Beispiel wäre ein MLM, das die Glucosewerte der Patienten zeitnah überprüft und gegebenenfalls einen Warnhinweis per SMS verschickt. Dessen grundlegendes Verhalten lässt sich folgendermaßen beschreiben:

WENN: Glucose < Grenzwert
DANN: Warnhinweis per SMS

Dieses MLM wird automatisch ausgelöst, sobald ein Glucosewert dokumentiert wird. Die automatisierte Überwachung klinischer Ereignisse, insbesondere der Dokumentation neuer Beobachtungen, bezeichnet man als Clinical Event Monitoring [6]. Die Arden-Syntax wurde gezielt auf dieses Einsatzgebiet zugeschnitten. MLMs weisen generell eine äußere Rahmenstruktur auf, die man sich wie ein Formular vorstellen kann. Abb. 1 zeigt links die grundlegende Rahmenstruktur eines MLMs. Es gibt Abschnitte (Kategorien), die in Felder (Slots) unterteilt sind. Das Verhalten eines MLMs zur Laufzeit wird dabei im Wesentlichen von vier Slots der Kategorie KNOWLEDGE gesteuert:

- EVOKE: Ereignisse, auf welche das MLM reagieren soll
- DATA: Die vom MLM benötigten Daten, z.B. die Glucosewerte des Patienten
- LOGIC: Die eigentliche medizinische Entscheidungslogik
- ACTION: Die ggf. auszuführende Aktion, z.B. eine SMS an den Arzt

Die von MLMs zu treffenden klinischen Entscheidungen werden mittels programmiersprachlicher Konstrukte innerhalb dieser Slots umgesetzt. Hierzu beinhaltet der Standard eine Reihe von Datentypen, Anweisungen und Operatoren, die denen aus bekannten Programmiersprachen ähneln, aber insbesondere durch eine betonte Nähe zur natürlichen Sprache und die bewusste Beschränkung auf einfache Konstrukte besonders leicht verständlich sind. Die Entscheidungslogik kann dabei je nach Anwendungsfall beliebig komplex werden. Abb. 1 zeigt rechts einige ausgewählte Details eines MLMs. Eine Nutzung nur der programmiersprachlichen Konstrukte ohne die umgebende Rahmenstruktur ist nicht möglich.

Abb. 1: Rahmenstruktur eines MLMs (links) sowie ausgewählte Details eines einfachen MLMs zum Monitoring von Glucosewerten (rechts).

MAINTENANCE	TITLE
	MLMNAME
	ARDEN
	VERSION
	INSTITUTION
	AUTHOR
	SPECIALIST
	DATE
	VALIDATION
LIBRARY	PURPOSE
	EXPLANATION
	KEYWORDS
	CITATIONS
	LINKS
KNOWLEDGE	TYPE
	DATA
	PRIORITY
	EVOKE
	LOGIC
	ACTION
	URGENCY

```
MAINTENANCE:
  ...
LIBRARY:
  ...
KNOWLEDGE:
    TYPE: data-driven ;;
    DATA:
        glucose := READ LATEST {<mapping>};
        ...
        ;;
    LOGIC:
        IF glucose IS LESS THAN limit THEN
            // Führe ACTION-Slot aus
            CONCLUDE TRUE;
        ENDIF;
        ;;
    ACTION:
        ...
        // "alert" ist ein Textbaustein,
        // "physicians" eine Telefonliste
        WRITE alert AT physicians;
        ;;
```

Einsatz der Arden-Syntax am UK Erlangen

Im Jahre 2006 begann auf den Intensivstationen des UKER die Umstellung von der papierbasierten auf die EDV-basierte Dokumentation in Form der Einführung eines kommerziellen Patientendatenmanagementsystems (PDMS). Heute ist dieses PDMS auf allen zwölf Intensivstationen des UKER mit etwa 170 Intensivbetten im Routineeinsatz. Gerade in der Intensivmedizin ist der Einsatz von MLMs besonders lohnenswert, da dort erheblich größere Datenmengen dokumentiert und überwacht werden müssen als auf Normalstationen. Im Rahmen einer Forschungskooperation mit dem PDMS-Hersteller wurde auf der interdisziplinären operativen Intensivstation der anästhesiologischen Klinik mit damals 25, heute 35 Betten eine kommerzielle Arden-Syntax-Ausführungsumgebung in das PDMS integriert [2].

Im Laufe dieses Integrationsprojektes wurde schnell ersichtlich, dass sich die Arden-Syntax über ihren eigentlichen Anwendungszweck hinaus in vielfältiger Weise nutzbringend einsetzen lässt. So entstand eine Reihe von MLMs zur Unterstützung der DRG-Kodierer bei der Abrechnung. Diese MLMs extrahieren die gewünschten Informationen auf Knopfdruck und ersparen den Kodierern das manuelle Sichten der Akten. Zudem erwiesen sich die programmiersprachlichen Konstrukte der Arden-Syntax als gut geeignet zur Implementierung von kompakten patientenübergreifende Listenansichten und klinischen Dashboards, die den klinischen Mitarbeitern das zeitraubende Wechseln zwischen einzelnen Akten möglichst ersparen. Im Gegensatz zu typischen MLMs, die automatisch starten, sobald ein bestimmtes Ereignis eingetreten ist, startet die Mehrzahl dieser MLMs auf Knopfdruck. Hierzu wurde ein MLM-Viewer an geeigneter Stelle in das PDMS integriert.

Die Bilanz der Erfahrungen

Die Erfahrungen mit der Arden-Syntax am UKER haben gezeigt, dass dieser Standard einen hohen Nutzwert in der klinischen Routine hat. Dabei entstanden in zunehmendem Maße MLMs, die deutlich vom typischen Einsatzgebiet des Standards abweichen. Insgesamt kann man sagen, dass wir die Arden-Syntax immer öfter als Ersatz für gängige Skriptsprachen verwendeten. Durch die für den medizinischen Einsatz optimierten Sprachkonstrukte ergab sich kompakter und leicht verständlicher Code. Ein weiterer Unterschied zu traditionellen MLMs bestand in der Darstellung der Ergebnisse. Während typische MLMs eine Kurzmeldung generieren, die kaum eine Formatierung erfordern, waren die MLMs am UKER mit einer teilweise recht aufwändigen Visualisierung verbunden. Man könnte sagen, dass die Arden-Syntax am UKER als eine Art App-Baukasten verwendet wurde und sich hierfür als geeignet erwies. Bei einer insgesamt deutlich positiven Erfahrungsbilanz zeigten sich aber auch einige Problembereiche des Standards. Zunächst stellte die Herstellung der technischen Interoperabilität mit dem PDMS einen erheblichen Aufwand dar. Dies lag vor allem an der Schnittstelle der damaligen PDMS-Versionen, aber auch von Seiten der Arden-Syntax gab es Einschränkungen beim Mapping und der Weiterverarbeitung von Patientendaten mit komplexer Struktur jenseits einfacher Listen oder Tabellen [7]. Zudem wurden formale Mängel in der Spezifikation des Standards ersichtlich, deren Publikation gerade in Arbeit ist. Ein weiterer Problembereich bestand darin, dass die Wiederverwendung von bereits geschriebenem Code zwar in gewissem Grad unterstützt wird, allerdings in einer Form, die in der Praxis umständlich zu handhaben ist. Auch zeigte sich, dass die äußere Rahmenstruktur nicht in allen

Fällen nützlich ist. Je weiter sich MLMs vom typischen Anwendungsspektrum entfernen, zum Beispiel bei Phenotyping [8] oder Arztbriefschreibung [9], desto mehr wird sie eher hinderlich.

Die experimentelle Weiterentwicklung

Aus der Bilanz der Erfahrungen mit dem Routineeinsatz der Arden-Syntax entstand eine experimentelle Weiterentwicklung, die nicht gezielt auf das Clinical Event Monitoring zugeschnitten ist, sondern darüber hinaus für verschiedene weitere Anwendungsgebiete innerhalb der Medizininformatik geeignet sein soll. Der Arbeitstitel für diese Sprache lautet **P**rogramming **L**anguage, **A**rden-**IN**spired (PLAIN). Der äußerlich auffälligste Unterschied zwischen der Arden-Syntax und PLAIN besteht darin, dass bei PLAIN die äußere Rahmenstruktur vollständig weggelassen werden kann. Zudem lassen sich zusätzliche alternative Rahmenstrukturen definieren. Während die Arden-Syntax bisher nur den Rahmentyp *data-driven* für das Clinical Event Monitoring unterstützt, gibt es bei PLAIN noch die Typen *sub-module* (klassische Modularisierung zur Wiederverwendung von Code) und *user-driven* (optimiert für MLMs auf Knopfdruck). Aktuell finden Forschungsarbeiten zu den Rahmentypen *dashboard* (patientenübergreifende Dashboard-Ansichten), *reporting* (deskriptive Statistik) und *phenotyping* (Patientenrekrutierung für Studien) statt.

Auf der Ebene der programmiersprachlichen Konstrukte entspricht der Prototyp von PLAIN zu etwa 80% der Arden-Syntax in Version 2.8, wobei einige neue Statements und Operatoren integriert wurden. Die weitere Entwicklung ist prinzipiell offen, der bewährte Standard stellt lediglich den Ausgangspunkt dar. Technisch betrachtet handelt es sich bei PLAIN um einen vollständigen Neuentwurf. Die Grammatik wurde neu geschrieben, um einige formale Unstimmigkeiten zu beseitigen und die zukünftige Integration experimenteller Sprachkonstrukte zu erleichtern. Der Prototyp besteht aus einem Compiler-Interpreter-Paar. Das Herzstück bildet eine virtuelle Maschine, die in der Programmiersprache Haxe geschrieben wurde und somit eine Übersetzung in diverse andere Sprachen unterstützt.

Hinsichtlich des Zugriffs auf Patientendaten besteht ein weiterer Unterschied zwischen der Arden-Syntax und PLAIN. Während der Standard das Datenmapping bisher vollständig der nutzenden Institution überlässt, wird es von PLAIN nativ unterstützt. Dazu wurde das Typsystem mit dem JSON-Datenformat verschmolzen sowie eine native Unterstützung von REST-Services integriert, kombiniert mit der Spezifikation eines Formates für externe Daten (wahlweise JSON oder XML). Während beim Einsatz der Arden-Syntax am UKER bisher die vom jeweiligen MLM benötigten Datenelemente separat gemappt wurden, gibt es bei PLAIN immer nur einen Mapping-Schritt, in dem komplexe Datenbestände, meist ganze Akten oder Listen ganzer Akten, auf eine kongruente Datenstruktur abgebildet werden.

Konsensfindung im Rahmen der Arbeitsgruppe Wissensbasierte Systeme

Bisher wurde PLAIN nur am Erlanger Lehrstuhl für medizinische Informatik für Forschung und Lehre sowie am UKER in der klinischen Routine eingesetzt. Ein früher Prototyp wurde im April 2017 auf der conhIT-Satellitenveranstaltung in Berlin vorgestellt. Im September 2017 wurde dann ein Workshop zu PLAIN auf der GMDS-Jahrestagung in Oldenburg abgehalten, um den Nutzerkreis um interessierte Standorte zu erwei-

tern, wobei eine kollaborative Weiterentwicklung angestrebt wird. Als Wegweiser sollen dabei die Erfahrungen und Bedürfnisse der tatsächlichen Anwender dienen. Die Konzertierung der Entwicklung soll im Rahmen der Arbeitsgruppe Wissensbasierte Systeme der GMDS erfolgen. Hierbei geht es vor allem um die Einführung zusätzlicher Leistungsmerkmale, insbesondere neuer Operatoren. Ein Ansatz für die Konsensfindung auf Basis etablierter Methoden wurde auf der GMDS-Jahrestagung im Rahmen der Arbeitsgruppe Wissensbasierte Systeme vorgeschlagen und diskutiert.

Ein App-Baukasten für klinische Informationssysteme?

Die in der klinischen Routine eingesetzten klinischen Arbeitsplatzsysteme sind meist kommerzieller Art. Das Leistungsspektrum solcher Systeme ist meist relativ statisch, und es ist keineswegs sichergestellt, dass es für den Kunden überhaupt eine Möglichkeit gibt, ein System durch selbst entwickelte Zusatzfunktionen zu erweitern. Sofern das Arbeitsplatzsystem allerdings die erforderlichen Features bereitstellt, wie eine Schnittstelle für den Zugriff auf Patientendaten, eine Ereigniserkennung und eingebettete Browser-Elemente, kann eine Institution das Leistungsspektrum durch die Integration von MLMs oder anderen Apps selbsttätig erweitern.

Die MLMs werden aber nur selten von den Klinikern entwickelt, sondern diese wenden sich in der Regel an diejenigen Kollegen aus der Medizininformatik, die aus der Kerninformatik stammen und in Programmierarbeiten geübt sind. Diese Kollegen haben aber normalerweise keine tiefergehenden Medizinkenntnisse, und hieraus resultiert ein in der Regel aufwändiger Abstimmungsprozess mit den Klinikern darüber, ob die angeforderte Zusatzfunktion auch richtig umgesetzt wurde. Aus dieser Sicht erscheint die Vorstellung verlockend, dass Kliniker die Zusatzfunktion selbsttätig erstellen könnten. In der Praxis kommt dies eher selten vor, denn nicht nur den Informatikern fehlen oft die Medizinkenntnisse, sondern umgekehrt sind bei Medizinern gute Programmierkenntnisse eher selten zu finden. Eine bewusst einfach konstruierte domänenspezifische Sprache könnte hier eine Brücke schlagen oder zumindest eine Kluft verringern. Dies war bereits bei der Arden-Syntax ein explizites Designziel [1]. Ein Teilziel von PLAIN in diesem Kontext ist die Erweiterung des Leistungsspektrums der Arden-Syntax um Visualisierungsfunktionen, mit denen Tabellen, Diagramme und Tooltips auf möglichst simple und verständliche Weise erstellt werden können. Dabei muss man sich natürlich darüber im Klaren sein, dass auch eine einfache domänenspezifische Programmiersprache einen nicht zu vernachlässigenden Einarbeitungsaufwand erfordert. Dieser ist aber deutlich geringer als bei einer Allzwecksprache wie Java oder C#. So wird beispielsweise die objektorientierte Programmierung sowohl von der Arden-Syntax als auch von PLAIN bewusst nicht unterstützt. Daher setzt deren Nutzung, im Gegensatz zu vielen gängigen Allwecksprachen, keine Kenntnisse der Objektorientierung voraus.

Vorverarbeitung klinischer Daten

Die Integration von Zusatzfunktionen in klinische Systeme erfordert oft eine Vorverarbeitung der Patientendaten. Ein häufiger Vorverarbeitungsschritt bezieht sich auf die Zeitstempel der klinischen Daten, denn diese passen nicht immer zusammen. Ein einfaches Beispiel ist der Horovitz-Quotient, bei dem der Zähler aus der BGA-Analyse, der Nenner aber von der Beatmungsmaschine stammt, was eine zeitfensterbasierte Vorverarbeitung erforderlich macht. Derartige Vorverarbeitungen von zeitgestempelten Daten werden in der Routine häufig benötigt, daher beinhaltet die Arden-Syntax ein zeitgestempeltes Typsystem und zahlreiche temporale Operatoren. Die Arden-Syntax hat die in der Praxis komfortable Eigenschaft, zusammenpassende Zeitstempel bei einfachen Berechnungen zu erhalten, d.h. sie automatisch dem Ergebnis zuzuweisen. Derzeit laufen Forschungsarbeiten, um diese Eigenschaft auch auf komplexere Berechnung auszudehnen.

Besonders bei der Vorverarbeitung mit ihren immer wiederkehrenden Arbeitsschritten kommt man mit der Problematik der Wiederverwendung von Code in Kontakt, die ausdrücklich kein Designziel der Arden-Syntax war. Dementsprechend gibt es hier deutlichen Optimierungsbedarf. So lässt sich beispielsweise eine zeitfensterbasierte Aggregation von zeitgestempelten Daten, die von vielen MLMs benötig wird, nicht generisch und vollständig wiederverwendbar implementieren, wodurch ein Kopieren von Codesegmenten nötig werden kann. Daher wurden PLAIN zwei experimentelle Sprachkonstrukte integriert, die diese Limitierung aufheben. Zudem wurde eine Möglichkeit integriert, MLMs als benutzerdefinierte Funktionen zu verwenden.

Fazit

Die Arden-Syntax ist ein bewährter, auf das Clinical Event Monitoring zugeschnittener Standard zur medizinischen Wissensrepräsentation, der unserer Erfahrung nach einen hohen praktischen Nutzwert aufweist und für viele zusätzliche Anwendungsgebiete eingesetzt werden kann. Die Erfahrung hat aber auch einige Problembereiche aufgezeigt, die eine experimentelle Weiterentwicklung namens PLAIN motiviert haben. Deren weitere Entwicklung soll kollaborativ im Rahmen der Arbeitsgruppe Wissensbasierte Systeme der GMDS erfolgen. ■

Literatur
[1] Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res. 1994;27:291–324.
[2] 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.
[3] Hripcsak G, Wigertz OB, Clayton PD. Origins of the Arden Syntax. Artif Intell Med 2015. doi:10.1016/j.artmed.2015.05.006.
[4] 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.
[5] Vetterlein T, Mandl H, Adlassnig K-P. Fuzzy Arden Syntax: A fuzzy programming language for medicine. Artif Intell Med. 2010;49:1–10. doi:10.1016/j.artmed.2010.01.003.
[6] Hripcsak G, Clayton PD, Jenders RA, Cimino JJ, Johnson SB. Design of a clinical event monitor. Comput Biomed Res. 1996;29:194–221.
[7] 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.
[8] Mate S, Castellanos I, Ganslandt T, Prokosch H-U, Kraus S. Standards-Based Procedural Phenotyping: The Arden Syntax on i2b2. Stud Health Technol Inform. 2017;243:37–41.
[9] 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.