

# Arden Syntax – An introduction

**Karsten Fehre**

Medexter Healthcare GmbH  
Borschkegasse 7/5  
A-1090 Vienna, Austria  
[www.medexter.com](http://www.medexter.com)

## Content

- What is Arden Syntax ?
- Arden Syntax – Fundamentals
- Basic MLM Layout
- Concepts: Data Types – Expressions – Statements – Operators
- Special Concepts: Curly Braces
- MLM to MLM Interaction
- Fuzzy Arden Syntax

## What is Arden Syntax ?

- ... a language used for representing and sharing medical knowledge.
- ... used for sharing computerized health knowledge bases across personnel, information systems, and institutions.
- ... organized in modules. Each module referred to as a medical logic module (MLM) contains sufficient knowledge to make a single decision.
- ... an executable format that can be used by clinical decision support systems.

## History

- A first draft of the standard was prepared at a meeting at the Arden Homestead, New York, in 1989. Arden Syntax was subsequently adopted as a standard by the American Society for Testing and Materials (ASTM) as document E 1460, under subcommittee E 31.15 Health Knowledge Representation.
- 1992: Arden Syntax version 1.0
- 1998: sponsorship moved to HL7 International (Arden Syntax Work Group)
- 1999: Arden Syntax version 2.0 approved by HL7 and the American National Standards Institute (ANSI)
- 2014: Arden Syntax version 2.10



## History

| Version | Year | Important changes  |
|---------|------|--|
| 2.1     | 2002 | new string operators; reserved word "currenttime" returns the system time  |
| 2.5     | 2005 | object capabilities: create and edit objects; XML representation of MLMs (except logic, action and data slot)  |
| 2.6     | 2007 | UNICODE encoding; additional resources category to define text resources for specific languages; time-of-day and day-of-week data types; "localized" operator to access texts in specific languages  |
| 2.7     | 2008 | enhanced assignment statement; extended "new" operator to allow easy and flexible object instantiation   |
| 2.8     | 2012 | additional operators for list manipulation; operators to manipulate parts of given date and time values; switch statements; keyword "breakloop" for aborting a loop; number of editorial corrections |
| 2.9     | 2013 | <b>fuzzification</b> : fuzzy data types and fuzzy sets; adjustment of all available operators to be able to handle fuzzy data types  |
| 2.10    | 2014 | <b>XML representation</b> of whole MLMs (including logic, action and data slot)  |

## Arden Syntax – Fundamentals I

- In Arden Syntax, medical knowledge is arranged within medical logic modules (MLMs)
- Each MLM represents sufficient knowledge to make a single clinical decision
- One or more MLMs are stored within a file that has the extension “.mlm”
- Each MLM is well organized and structured into categories and slots with specific content

```
maintenance:
  title:      [TITLE_(needed)];;
  mlmname:   [MLM-NAME_(needed)];;
  arden:     Version 2.5;;
  version:   [MLM-VERSION_(needed)];;
  institution: [INSTITUTION_(needed)];;
  author:    ;;
  specialist: ;;
  date:     [DATE];;
  validation: testing;;

library:
  purpose:    ;;
  explanation: ;;
  keywords:  ;;
  citations:  ;;
  links:     ;;

knowledge:
  type:      data_driven;;
  data:     ;;
  priority:  ;;
  evoke:    ;;
  logic:
    conclude true;
  ;;
  action:
  ;;
  urgency:  ;;
end:
```

## Arden Syntax – Fundamentals II

- MLMs are working in close contact with their host system. Ways of interaction are:
  - **Input:** By calling an MLM, an input parameter can be committed
  - **Curly Brace Expressions:** So called “curly brace expressions” implement a special kind of dynamic interaction between MLMs and host systems
  - **Write Statements:** Texts can be written to destinations that are maintained by the host system
  - **Output:** Analogous to the input parameter, data can be committed from the MLM to the host system after the execution of the MLM has finished
- In order to start the execution of an MLM, an engine is needed that handles communication with the host system and can tell which of the MLMs is available
- Ways to start running an MLM:
  - **MLM call:** An MLM is directly called
  - **Event call:** Any MLM that listens to a specific event is executed

## Basic MLM Layout

- An MLM is composed of slots, grouped into the following **four** required **categories**: **maintenance**, **library**, **knowledge**, and **resources**
- Categories must appear in the correct (predefined) order
- Within each category is a set of **slots** that must appear in the correct order, too
- In general, an MLM is arranged such as:

```

maintenance:
  slotname: slot-body;;
  ...
library:
  slotname: slot-body;;
  ...
knowledge:
  slotname: slot-body;;
  ...
resources: <optional>
  slotname: slot-body;;
  
```

The image shows a vertical stack of four colored boxes, each representing a category in an MLM. The top box is light blue and labeled 'maintenance'. The second box is light green and labeled 'library'. The third box is light yellow and labeled 'knowledge'. The bottom box is light purple and labeled 'resources'. Each box contains a large title and a snippet of code. The code includes comments and instructions for how to use the slots in each category. For example, in the 'maintenance' section, it says 'The app is not used when you, PHYSICIAN, receive a message containing the SMN and the classification will be zero.' In the 'library' section, it says 'The app is not used when you, PHYSICIAN, receive a message containing the SMN and the classification will be zero.' In the 'knowledge' section, it says 'The app is not used when you, PHYSICIAN, receive a message containing the SMN and the classification will be zero.' In the 'resources' section, it says 'The app is not used when you, PHYSICIAN, receive a message containing the SMN and the classification will be zero.'



## Maintenance Category

- Contains slots that specify general information, unrelated to the MLM's medical knowledge
- These slots are used for MLM knowledge base maintenance and version control
- Contains information about the Arden Syntax version in use
- Slots:
  - Title
  - MLMname (content required)
  - Arden Syntax version (content required)
  - Version (content required)
  - Institution (content required)
  - Author
  - Specialist
  - Date
  - Validation (possibilities: production, research, testing, expired)
- MLMname, Institution, and Version are used to identify the MLM



## Library Category

- Contains the slots pertinent to knowledge base maintenance that are related to the MLM's medical knowledge
- Slots provide health personnel with explanatory information as well as links to relevant health literature
- Slots
  - Purpose
  - Explanation
  - Keywords
  - Citations
  - Links



## Knowledge Category

- Contains the slots that actually specify the MLM's action
- This category's slots define:
  - Terms used in the MLM (data slot)
  - The order of execution if more than one MLM is called (priority and urgency slot)
  - The context in which the MLM should be evoked (evoke slot)
  - The condition to be tested (logic slot)
  - The action to be taken if the condition is true (action slot)
- Slots:
  - **Type:** only "data-driven" available
  - **Data:** preparation, preprocessing, and query of data
  - **Priority:** defines the order of MLM evaluation
  - **Evoke:** checks if the MLM has to be executed if an event call occurs
  - **Logic:** contains the actual logic of the MLM
  - **Action:** is executed if logic slot concludes **true**
  - **Urgency:** defines the urgency of the action slot



## Resources Category

- Contains a set of language slots that specify the textual resources from which the localized operator should draw to obtain message content in different languages
- Each language slot defines a set of key/value pairs that represent text constants in one specific language
- At least one language slot is required
- Slots:
  - Default (defines the default language to be used)
  - Language (one language slot for each language to be used)
- Example:

```
resources:
  default: de;;
  language: de
    'msg' : "Der BMI %.1f des Patienten ist nicht im Normalbereich und wird
    bewertet als ";
  ;;
  language: en
    'msg' : "The patient's BMI %.1f is not in the normal range and is
    classified as ";
  ;;
```



## Sample MLM

- Most of the examples for operator and concept explanation are taken from the following sample MLM which calculates the body mass index (BMI) of a patient:

```
maintenance:
  title:          body mass index;;
  mlmname:       BMI_HowTo;;
  arden:         Version 2.9;;
  version:       1.00;;
  institution:   Medexter Healthcare;;
  author:        ;;
  specialist:    ;;
  date:         2016-05-09;;
  validation:    testing;;
library:
  purpose:       body mass index;;
  explanation:   calculation of body mass index;;
  keywords:     BMI, body mass index;;
  citations:     ;;
  links:        en.wikipedia.org/wiki/Body_mass_index;;
```

## Sample MLM (cont.)

```
knowledge:
  type: data_driven;;
  data:

    // MLM that contains the interface definition "LET get_birth BE INTERFACE {Patient.dateOfBirth}"
    mlmImport      := MLM 'interface_birthday_definition';

    // include
    include mlmImport;

    mlmForReadSize := MLM 'read_Size_MLM'; // MLM that can read the current size of the patient from DB

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

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

    // read all measured weights from the database
    LET weights    BE READ {SELECT measured_weight FROM DB WHERE patID = patientID};

    LET userEvent  BE EVENT {getBMI};

    // object declaration
    bmiResult     := object [bmi, classification];
  ;;
  priority:      ;;
  evoke:
    userEvent;
  ;;
```

## Sample MLM (cont.)

```
logic:
  result := new bmiResult; // create an empty result object

  weight := latest of weights; // get the latest weight from the list

  size := call mlmForReadSize with patientID; // get the size of the patient calculated by another MLM

  // calculation of BMI
  let result.bmi be weight / (size ** 2); // calculation of BMI
  age := currenttime - birth; // calculation of AGE

  // Classification - the classification is only valid for patients older than 19
  if the age is less than 19 years then result.classification := null;
  elseif the result.bmi is less than 18.5 then result.classification := localized 'under';
  elseif the result.bmi is less than 25 then result.classification := null;
  else let the result.classification be localized 'over';
  endif;

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

  if (time of weight) is before (currenttime - 6 months) then
    conclude false; // no bmi calculation if the latest measure was 6 months ago
  else
    conclude result.classification is present; // if there is a classification, execute the action slot
  endif;
```

## Sample MLM (cont.)

```
action:
  write result.bmi || result.classification || "."; // write result
  return result; // return result object
;;
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:
```