MEDINFO 2004
M. Fieschi et al. (Eds)
Amsterdam: IOS Press
© 2004 IMIA. All rights reserved

Description and Status Update on GELLO: a Proposed Standardized Object-Oriented Expression Language for Clinical Decision Support

Margarita Sordo, Aziz A. Boxwala,Omolola Ogunyemi, Robert A. Greenes

Decision Systems Group, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA

Abstract

A major obstacle to sharing computable clinical knowledge is the lack of a common language for specifying expressions and criteria. Such a language could be used to specify decision criteria, formulae, and constraints on data and action. Although the Arden Syntax addresses this problem for clinical rules, its generalization to HL7's object-oriented data model is limited.

The GELLO Expression language is an object-oriented language used for expressing logical conditions and computations in the GLIF3 (GuideLine Interchange Format, v. 3) guideline modeling language. It has been further developed under the auspices of the HL7 Clinical Decision Support Technical Committee, as a proposed HL7 standard., GELLO is based on the Object Constraint Language (OCL), because it is vendor-independent, object-oriented, and side-effect-free. GELLO expects an object-oriented data model. Although choice of model is arbitrary, standardization is facilitated by ensuring that the data model is compatible with the HL7 Reference Information Model (RIM).

Keywords:

Expression language, clinical decision support.

Introduction

Arden Syntax is the only current standardized approach to representation of decision logic, used in some health care information systems for encoding of Medical Logic Modules (MLMs), or single-step *if...then* rules [1]. However, Arden Syntax has a number of well-recognized limitations, primarily in its data model. It uses a simple atomic time-stamped, non-object-oriented data model. Arden Syntax is endorsed as a standard by HL7, but it is not compatible with the object-oriented data model of the proposed HL7 Reference Information Model v. 3 [2]. It has limited ability to express temporal conditions, and is not readily extensible.

A major obstacle to sharing clinical knowledge in Arden Syntax is the lack of a common format for data encoding and manipulation. Although the Arden Syntax addresses this problem by isolating references to local data in curly braces ["{}"] in MLMs, it does not provide mechanisms for accessing data in a truly format-independent way. Each MLM must specify how data elements used in the logic are obtained from the host environment within curly braces in its data section [3].

The Decision Systems Group (DSG) has been working in the area of decision support for many years. Coauthors of this study have worked with colleagues at Stanford and Columbia in the InterMed project, aimed at developing a standard approach to representing clinical guidelines known as GLIF (GuideLine Interchange Format) [4,5]. Based on InterMed's request to HL7, in 2000, a Clinical Guidelines Special Interest Group (CG SIG) was established to pursue standardization of guideline modeling, and was made part of a Clinical Decision Support Technical Committee (CDS TC), with Arden Syntax (already part of HL7) becoming a second SIG under the CDS TC. In the CDS TC, DSG participants have focused on refining the expression language for query statements and decision rules used in GLIF v.3, known as GELLO. GELLO is being considered by HL7 as a possible successor to the expression language in Arden and as a basis for query and rule expressions for guidelines. We expect that GELLO will be ready for balloting as a standard in May 2004 [6].

Motivation and Background

The main motivations for developing the GELLO expression language are the need for a standard, object-oriented language to formulate expressions for manipulating clinical data and providing decision support in various clinical applications, and the desirability of sharing the clinical knowledge expressed in the language. Ideally, such a language should be: vendor independent, platform independent, object oriented and compatible with the HL7 Reference Information Model (RIM), easy to read and write, side-effect free, extensible and shareable.

GELLO is intended to include both query and expression sublanguages, which we refer to collectively as the GELLO language. GELLO is based on the Object Constraint Language (OCL) [7], developed by the Object Management Group. OCL is being used by other HL7 TCs for specifying model constraints and GELLO, which is derived from OCL, is of interest to other parts of HL7 such as the Compound Document Architecture TC, the Templates TC, and the Controlled Query TC.

OCL was developed to add precision to the diagrammatic elements of the Unified Modeling Language (UML). OCL allows modelers to express all the relevant aspects of a specification that UML diagrams by themselves cannot represent. OCL is a strongly-typed, object-oriented, pure expression language without side effects. Although the evaluation of an OCL expression returns a value, the state of the underlying data model cannot change because of the evaluation of such an expression. OCL is the result of a consensus effort towards a standard in object-oriented modeling and design. It does not rely on a specific platform.

Relevant components of OCL have been selected and integrated into GELLO query and expression languages to provide a suitable framework for manipulation of clinical data for decision support. This paper focuses on the expression sublanguage. The GELLO expression language can be used to:

- Construct decision criteria by building up expressions to reason about particular data features/values. These criteria can be used in decision-support knowledge bases such as guidelines.
- Create expressions, and formulae in a variety of applications.

The object-oriented approach using a general purpose language has the flexibility and extensibility needed for implementation in a broad range of applications. By basing GELLO on OCL, it has the vendor and platform-independence and side-effect-free qualities of OCL. To make the language suitable for clinical decision support:

- GELLO provides mechanisms to access clinical data through an OO data model (the HL7 RIM), with strongly-typed expressions.
- All data manipulation methods must be explicitly defined in the OO data model.
- By using a specified OO data model, such as the HL7 RIM, each guideline rule, or template constraint need not provide a separate mechanism for translation of data elements to/from host environments (as in the "curly braces" needed in the Arden Syntax data section). See Fig. 1.

HL7 RIM

The clinical part of the HL7 RIM models medical knowledge and patient data. All clinical data are specified as *Act* objects or *Acts*. Acts have attributes that provide information about clinical concepts they represent and their timing. Relationships are used to model the Act circumstances and allow grouping the Acts and reasoning about them. Acts are specialized into procedures performed on a patient, observations about the patient, medications given to a patient, and other kinds of services. Different subclasses in the RIM contain additional attributes that help characterize an Act. For example, Observation has a value attribute, whereas Medication has attributes about dosage and route of administration. Act objects have a mood that distinguishes the ways in which they can be conceived as an event that occurred, a definition, an intent, order, etc.

vMR

Because the HL7 RIM is quite complex, the concept of a "virtual medical record" (vMR) [8], as a simplification of the HL7 RIM has been proposed for use in decision support. Currently still under development in the HL7 CDS TC, the vMR is itself an object-oriented data model.

UML ITS Data Types

The UML Implementable Technology Specification (ITS) is a collection of data types based largely on the proposed UML2 specification and it implements the functionality of the abstract data types specification of the HL7 RIM. It helps in mapping HL7 data types into core UML and OCL kernel data types and resolving inconsistencies where necessary. It provides a formally correct UML declaration of the HL7 data types [9]. Within the context of our implementation of an expression language, the UML ITS helps in mapping 'unknown', the third Boolean value in our logic, into the HL7 'null' value, used as both a third Boolean value and as a data value.

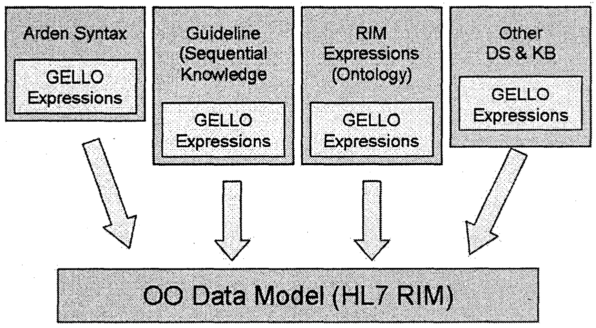

Figure 1 - GELLO Expressions in Different Applications Sharing a Common OO Data Model

GELLO Expression Language

GELLO's expression language is strongly typed and object-oriented. In order to facilitate the process of encoding and evaluation of expressions and more importantly, to maximize the shareability of queries and expressions, GELLO provides basic built-in data types, while providing the necessary mechanisms to access an underlying data model with all its associated classes and attributes.

Although this approach represents a paradigm shift in data representation, moving from time-stamped atomic data types to an object-oriented data model, it provides a first approach towards a standard for exchange, management and integration of clinical data.

Basic Data Types, Model Classes, Tuple and Collection Data Types

By providing only the basic data types with a restricted usage, with the necessary mechanisms for creating instances of classes in the data model, the GELLO expression language preserves portability, and extensibility, since additional classes can be incorporated into the data model for added functionality.

GELLO supports all basic data types: integer, real, string and Boolean which may be used to declare literals, and the tuple and collection types, inherited from OCL.

GELLO provides the notation for accessing classes and their properties in an underlying data model (e.g., HL7 RIM, or a vMR).

A GELLO collection is an abstract type with concrete collection types as subtypes. As in OCL, a GELLO collection has three types: Set, Bag and Sequence.

- A **Set** is the mathematical set. It does not contain any duplicate elements. All elements have the same type.
- A **Bag** is a collection of elements. Duplicates are allowed. All elements in a Bag have the same type.
- A **Sequence** is a collection with ordered elements. Duplicates are allowed. All elements in a Sequence must have the same type.

Inherited from the latest version of OCL, a *tuple* combines elements with different types into an aggregate type. Each tuple part has a name and a type. A tuple part can be a single element or a collection. The type of a tuple part can be of a basic or a model type. The syntax of a tuple is as follows:

Tuple{ $label_1$::$value_1$, ..., $label_n$: $value_n$},

Where $label_i$ is the label of the element *ith* and $value_i$ is a valid value – there is a valid data type (basic or model) that can hold such a value. Tuples may be used as a return type from queries that retrieve information from more than one data source.

Syntax

The style and notation used in GELLO conform to OCL's. GELLO syntax for the expression language is defined in Backus Naur Form (BNF) [6]. The general notation is object-oriented, that is, objects have properties. Such properties can either be attributes and/or operations and methods that can be accessed by the proper notation: an object name followed by a dot and the name of the property: *object.property.*

- **Attributes.** The value and type of an expression that refers to an attribute of an object are the type and value of the attribute to which such an expression refers.
- **Operations and Methods.** Operations and methods are operations associated to a class. Such operations may have parameters that must be included in the operation or method call. For example, ***aPerson.income(aDate)*** *aDate* is passed as parameter to the method *income* associated to the object *aPerson.* The type of the result is the return type of the operation.

Variables

Although GELLO has been conceived as a side-effect-free language, it provides facilities so that temporary, scope-delimited variables can be declared to hold data. A variable declaration declares a variable name and binds it to a type. The type of a variable must be specified. It can be either a basic or a model type. In either case, a variable can only hold a value of the same type.

In GELLO, variables can be declared using the *let* OCL expression. The syntax is as follows:

Let VarName: Type= initExpression

Where *initExpression* is the initial value of the variable. If the variable is of model type, the *initExpression* must include the *new()* operator as in:

Class.new(parameters)

Where *Class* is a class in the data model, *new* is the associated method for creating a new instance of a class *Class*, and *parameters* are the values for the class attributes. Some examples:

- let threshold_for_osteodystrophy : integer = 70
- let potassium : PhysicalQuantity = PhysicalQuantity.new(70, 'dl')

Operators

GELLO provides a full complement of arithmetic, Boolean, tuple, collection and string operators. GELLO incorporates from OCL standard operations to handle elements in collections. These operations take the elements in a collection and evaluate a GELLO expression on each of them: *select, reject, collect, forAll, iterate, exists* and *flatten.* The full description of the operators can be found in [6].

Expressions

A GELLO expression is any text string conforming to the definition of an expression in the GELLO language specification. When an expression is evaluated, the result of such evaluation is a value. The type of the result is the type of the expression.

Evaluation of an expression does not produce any side effects, although the returning value can be used by the guideline, rule, or embedding application to make decisions, control execution flow, etc. If an expression can be embedded in a conditional statement, the returning value is interpreted by the application to which the conditional statement belongs.

If an expression denotes a variable or a value, then such an expression has a type that must be checked for compatibility. Such a variable or value must match any of GELLO data types, or classes defined in the underlying data model.

If a value is assigned to a variable, both the returning value and the variable to which it is assigned must be of the same type.

Expression Evaluation

Expressions are evaluated from left to right. In the case of infix operators, the evaluation order is determined by the precedence of the operators. Argument lists included in method invocations are evaluated left-to-right.

Examples of Expressions

When the following expressions are evaluated, they return a value of type Boolean. Expressions like these can be used to build decision criteria:

- calcium.notEmpty() and phosphate.notEmpty()
- renal_failure and calcium_phosphate_product > threshold_for_osteodystrophy

GELLO and Decision Support Applications

We see an important potential for the GELLO expression language as a means for sharing clinical knowledge within applications. As depicted in Figure 1, GELLO expressions can be embedded in multiple applications. Further, since GELLO expressions refer to a common OO data model, they can be shared through knowledge repositories and directly within applications. However, GELLO is not constrained to a specific object model. Hence, potential object models not only provide data models, but also provide methods for reasoning and computation. In other words, GELLO relies on the general concepts of an Object-Oriented model but not on a specific data model.

```
let renal_failure_threshold :=
    PhysicalQuantity.new(2.0, "mg/dl")
let renal_failure :=
if creatinine->notEmpty() and
  creatine.observed_quantity() > renal_failure_threshold
then
  true
else
  false
endif
let threshold_for_osteodystrophy := 70
calcium_phosphate_product :=
  if calcium->notEmpty() and phosphate->notEmpty() then
    calcium.observed_quantity.value *
      phospate.observed_quantity.value
  else
    -1
  endif
```

Figure 2 - GELLO Expressions in Arden MLM data slot

GELLO could be used for encoding expressions and statements in Arden Syntax's MLM data and knowledge slots. The example in Figure 2 shows an excerpt of an MLM data slot. Expressions in **bold** represent GELLO expressions embedded in the Arden application. Such expressions refer to HL7 RIM classes e.g., PhysicalQuantity. While we could encode much of the platform-specific code in the data slot as expressions in GELLO, in some instances this could not be done because of limitations of the VMR, hence the direct reference to the HL7 RIM.

Discussion

GELLO is intended to facilitate sharing of medical knowledge that references a common data model. For standardization, the HL7 RIM is the target data model, or subsets of the RIM that are easier for developers to comprehend and manipulate can be constructed for specific purposes, such as the vMR, intended for decision support applications.

GELLO has been used by us to construct a number of example decision support rules, and an interpreter is under construction. Integration of query capabilities has been included in the latest specification of the language [6]. The HL7 CDS TC has proposed to ballot the specification of GELLO query and expression language during the upcoming meeting in May 2004.

Because the HL7 RIM is object-oriented, the transition to an object-oriented expression language that uses an object-oriented data model is desirable. Nonetheless, it represents a paradigm shift for those developers accustomed to Arden Syntax or other non-object-oriented approaches.

We are at the early stages of both adoption of the RIM and development of languages and tools to manipulate it. GELLO development has been occurring at a time when many other standards are in the process of development, including OCL. Thus experience will be necessary to determine directions for further evolution.

Our own efforts are focused on analyzing decision support resources in our own health care institution (Partners HealthCare System), and exploring how the knowledge used in the variety of applications which provide decision support can be better managed [10]. We expect that this analysis will provide an opportunity to evaluate GELLO as a means for encoding the knowledge.

Acknowledgments

We appreciate the helpful insights and constructive comments of Grahame Grieve, Gunther Schadow, and members of the HL7 CDS TC.

References

[1] Hripcsak G, Cimino JJ, Johnson SB, Clayton PD. The Columbia-Presbyterian Medical Center decision-support system as a model for implementing the Arden Syntax. *Proc Ann Symp Comput Appl Med Care.* 1991;248-252

[2] HL7 RIM http://www.hl7.org/Library/data- model/RIM/ modelpage_mem.htm

[3] Jenders RA, Sujansky W, Broverman CA, Chadwick M. Towards improved knowledge sharing: assessment of the HL7 Reference Information Model to support medical logic module queries. *Proc AMIA Ann Fall Symp.* 1997;308-312.

[4] Ohno-Machado L, Gennari JH, Murphy SN, Jain NL, Tu SW, Oliver DE, Pattison-Gordon E, Greenes RA, Shortliffe EH, Barnett GO. The guideline interchange format: a model for representing guidelines. *J Am Med Inform Assoc.* 1998 Jul-Aug;5(4):357-372.

[5] Peleg M, Boxwala AA, Ogunyemi O, Zeng Q, Tu S, Lacson R, Bernstam E, Ash N, Mork P, Ohno-Machado L, Shortliffe EH, Greenes RA. GLIF3: the evolution of a guideline representation format. *Proc AMIA Symp.* 2000; 645-649.

[6] Sordo M, Ogunyemi O, Boxwala AA, Greenes RA. Software Specifications for GELLO: An object-oriented query and expression language for clinical decision support. DSG Technical Report DSG-TR-2003-02, http:// www.hl7.org/Library/Committees/dss/ GELLOspecsJan041%2EPDF 2004.

[7] OCL http://www.klasse.nl/ocl/subm-draft.html

[8] The Virtual Medical Record (vMR) http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-2001-0876.pdf, 2001

[9] UML ITS data model. http://www.hl7.org/v3ballot/html/ foundationdocuments/itsuml/datatypes-its-uml.htm

[10]Greenes RA, Sordo M, Zaccagnini D, Meyer M, Kuperman GJ. Design of a Standards-Based External Rules Engine for

Decision Support in a Variety of Application Contexts: Report of a Feasibility Study at Partners HealthCare System. To appear in *Medinfo Proceedings*, 2004.

Address for correspondence

Margarita Sordo, D.Phil.
Decision Systems Group, Brigham and Women's Hospital,
Harvard Medical School, Boston MA.
msordo@dsg.harvard.edu