Improving the Usability of HL7 Information Models by Automatic Filtering

Antonio Villegas and Antoni Olivé
Services and Information Systems Engineering Department
Universitat Politècnica de Catalunya
Barcelona, Spain
Email: {avillegas, olive}@essi.upc.edu

Josep Vilalta
HL7 Education & e-Learning Services
HL7 Spain (Health Level Seven International)
Barcelona, Spain
Email: jvilalta@vico.org

***Abstract*—The amount of knowledge represented in the Health Level 7 International (HL7) information models is very large. The sheer size of those models makes them very useful for the communities for which they are developed. However, the size of the models and their overall organization makes it difficult to manually extract knowledge from them.**

We propose to extract that knowledge by using a novel filtering method that we have developed. Our method is based on the concept of class interest as a combination of class importance and class closeness. The application of our method automatically obtains a filtered information model of the whole HL7 models according to the user preferences. We show that the use of a prototype tool that implements that method and produces such filtered model improves the usability of the HL7 models due to its high precision and low computational time.

***Keywords*-Usability, Health Level Seven International, HL7, Models, Filtering, UML**

I. Introduction

The Health Level Seven International (HL7) is a not-for-profit, ANSI-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services [1].

HL7 develops specifications, the most widely used being a messaging standard that enables disparate healthcare applications to exchange key sets of clinical and administrative data. The HL7 standard specifications are unified by shared reference models of the healthcare and technical domains [2], [3].

The amount of knowledge represented in the HL7 information models is very large and continuously improved. The sheer size of those models makes them very useful to the communities for which they were developed: HL7 international affiliates with more than fifty HL7 active working groups (Structured Documents, Clinical Decision Support, Clinical Genomics...), large integrated healthcare delivery networks, government agencies and other organizations that use those models for the development of their enterprise information architecture of health systems [4], [5].

However, the size of HL7 information models and their organization makes it very difficult for those communities to manually extract knowledge from them. This problem is shared by other large models [6].

Currently, there is a lack of computer support to make those models usable for the goal of knowledge extraction. In this paper, we propose to extract that knowledge by using a novel filtering method that we have developed, and we show that the use of our prototype implementation of that method improves the usability of HL7 information models.

The structure of the paper is as follows. Section II introduces the HL7 models and describes the main UML constructs used to build them. Section III describes the concept of class importance and references the methods that can be used to compute it. Section IV describes the concept of class interest with respect to a filter set of classes and explains how to compute it. Section V presents our model filtering method. Section VI evaluates the use of the method in the context of the HL7 models. Finally, Section VII summarizes the conclusions and points out future work.

II. HL7 Information Models

Types of Models

The HL7 information models comprise three types of models. Each of the model types is based on the UML, although the concrete notation used differs depending on the model type. Also, the models differ from each other in terms of their information content, scope, and intended use. The following types of information models are defined:

- *Reference Information Model (RIM)* - The RIM is the information model that encompasses the HL7 domain of interest as a whole. The RIM is a coherent, shared information model that is the source for the data content of all HL7 interoperability artifacts: V2.x messages and XML clinical documents CDA R2 [3].
- *Domain Message Information Model (D-MIM)* - A D-MIM is a refined subset of the RIM that includes a set of classes, attributes and relationships that can be used to create messages and structured clinical documents for a particular domain (a particular area of interest in healthcare). There are predefined D-MIMs for a set of over 15 universal domains, such as Accounting and Billing, Care Provision, Claims and Reimbursement, and so on.

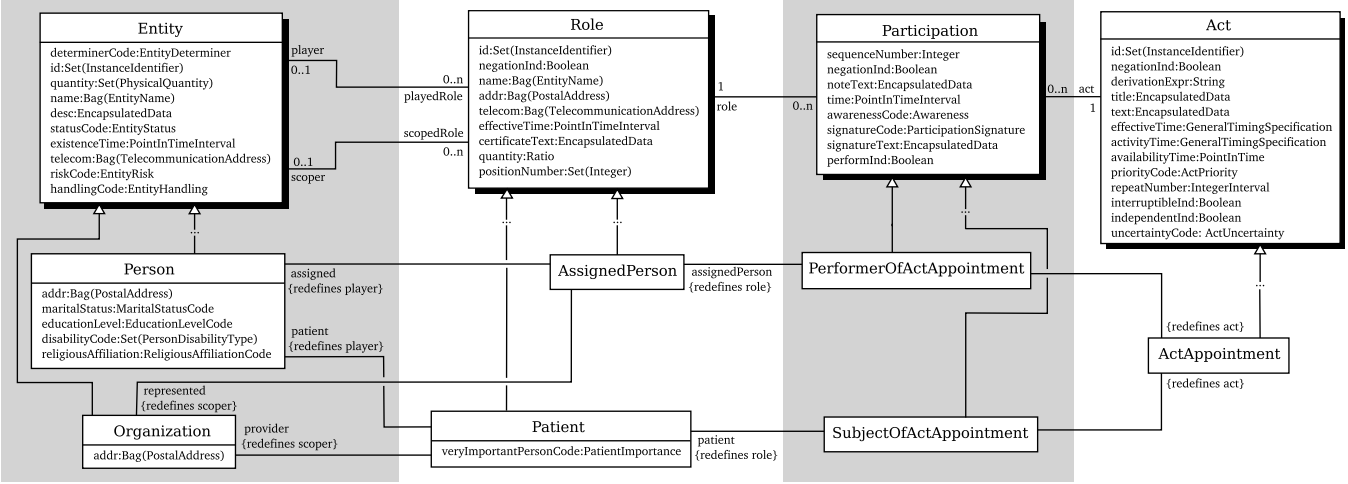

Figure 1. Sample of HL7 RIM refinements related to *ActAppointment* class

- *Refined Message Information Model (R-MIM)* - The R-MIM is a subset of a D-MIM that is used to express the information content for a message/document or set of messages/documents with annotations and refinements that are message/document specific. The content of an R-MIM is drawn from the D-MIM for the specific domain in which the R-MIM is used.

Structure of the HL7 Information Models

The RIM, D-MIM and R-MIM models can be analyzed as if they were built using in a particular way a small subset of constructs provided by the UML [7]. Figure 1 illustrates with a very small fragment of the RIM and of one D-MIM the main UML constructs used. RIM comprises six backbone classes: *Act*, *Participation*, *Entity*, *Role*, *ActRelationship* and *RoleLink*. Figure 1 shows the first four of these classes. Each one has a number of attributes with a defined multiplicity. Surprisingly, there are only eight main associations between the RIM classes, all of them binary and with their corresponding multiplicities. Figure 1 shows four of these associations.

Each of the RIM classes has many subclasses, although only a few of them are explicitly shown in the diagrams of the HL7 RIM specification. There are many specialization/generalization relationships (called *IsA* relationships, e.g. *Organization IsA Entity*) in the HL7 models. The number of RIM classes and subclasses is over 2,500. Figure 1 shows seven subclasses of four of the backbone RIM classes and seven *IsA* relationships.

D-MIM models refine the RIM in three ways:

1) The participants of one of the eight main associations defined between RIM classes are refined in the subclasses. This is the refinement most often used in the HL7 models. Note that it is not allowed to add new associations.
2) The multiplicities of an association defined between RIM classes are strengthened in the subclasses.
3) The multiplicity of an attribute of a RIM class is strengthened in a subclass. An optional attribute in a RIM class can be made mandatory or not allowed in a subclass. Note that it is not allowed to add new attributes.

R-MIM models refine D-MIM models in the same way. In all cases, the three kind of refinements can be expressed using UML constructs.

Figure 1 shows a few refinements related to the *ActAppointment* class. The instances of this class are appointments (a particular kind of *Act*). There may be several kinds of participations in an appointment. Figure 1 shows only two of them: *PerformerOfActAppointment* and *SubjectOfActAppointment*. To indicate that when the act is an appointment then the participations must be instances of *PerformerOfActAppointment* or of *SubjectOfActAppointment*, we redefine the association *Participation-Act* as shown in the figure. Note that *redefinition* is a UML construct, which is very useful in situations like this one. The redefinition of the association *Role-Participation* is similar. The overall semantics of these redefinitions is that the performer of an appointment is a *Person* that plays the role *AssignedPerson*, and that the subject of an appointment is a *Person* that plays the role *Patient*.

Sometimes, the UML redefinition construct does not allow the graphical representation of the strengthening of association multiplicities. In these cases, the redefinition must be formally captured by OCL invariants. For example, in Figure 1, the refinement of act in *SubjectOfActAppointment* also implies that an instance of *ActAppointment* is associated with a non-empty set (1..*) of *SubjectOfActAppointment*. However, this cannot be expressed graphically, and an OCL invariant must be used instead [8].

Figure 1 also shows the redefinitions of associations *player-playedRole* and *scoper-scopedRole* between *Entity* and *Role*. The *player* and the *scoper* of an *AssignedPerson* and of *Patient* must be a *Person* and an *Organization*, respectively.

III. Importance of HL7 Classes

Our filtering method is based on the concept of class importance. The importance of a class is a real number that measures the relative importance of that class in a model. We will see in the next section that we use that importance to select which classes are shown to the users.

There exist different kinds of methods to compute the importance of classes in the literature. The simplest family of methods is that based on *occurrence counting* [9]–[11], where the importance of a class is equal to the number of characteristics the class has represented in the model. These methods are class centered in the sense that the importance of a class depends only in the information the class has. Therefore, the more information about a class, the more important it will be.

Another family of methods are those based in *link analysis* [11], [12], where the importance of a class is defined as a combination of the importance of the classes that are connected to it with associations and/or *IsA* relationships. Such recursive definition results in an equation system and indicates that the more important the classes connected to a class are, the more important such class will be. In these methods the importance is shared through connections, changing from a class centered philosophy to a more interconnected approach of the importance. Iterative methods are required to solve the importance equation system, which increases the computational cost of this kind of methods.

Finally, there are some methods that even use the information about the existing instances of the classes and the associations of the model. Therefore, the importance they compute takes into account the structural part of the model but also the data that the classes instantiate. The problem with this family of *instance-dependent* methods [13], [14] is that without instances the method cannot be used.

As an example, Table I shows the 10 most important classes of the HL7 models[1] computed using the CEntityRank importance algorithm (see 3.6 of [15]). To compute this importance, the method takes into account the classes, the *IsA* relationships between them, the attributes and their multiplicities, the associations and their multiplicities, the association redefinitions and the OCL invariants.

The in-depth study of the computation of the importance of classes is beyond the scope of this paper. A review of methods to compute the importance taking into account different levels of knowledge is given in [15].

[1]The results have been obtained taking into account the RIM and the following D-MIM models: Laboratory, Account and Billing, Scheduling and Medical Records [1].

Table I
Top-10 Most Important Classes.

Rank	Class	Importance
1	Act	7.51
2	Role	5.11
3	ActRelationship	4.03
4	Participation	3.67
5	Entity	3.5
6	Observation	2.64
7	InfrastructureRoot	1.81
8	Organization	1.72
9	RoleLink	1.59
10	FinancialTransaction	1.54

The filtering method described in the next sections can be used in connection with any of the existing methods for computing the importance of classes.

IV. Interest of HL7 Classes

The importance of a class is an absolute metric that depends only on the whole set of HL7 models. The metric is useful when a user wants to know which are the most important classes, but it is of little use when the user is interested in a specific subset of classes, independently from their importance. What is needed then is a metric that measures the interest of a class with respect to such set, that we call filter set.

A filter set $\mathcal{FS}$ of classes is a non-empty set of classes from the HL7 models. The filter set comprises the minimum set of classes in which a user is interested at a particular moment. For example, if the user wants to see what is the knowledge the models have about classes *Patient* and *ActAppointment*, then she defines $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$. We will see in the next section that starting from this filter set, our filtering method retrieves the knowledge represented in the models about *Patient* and *ActAppointment* that is likely to be of more interest to the user.

Additionally, it is possible to define a set of classes not to be considered in the filtering method. We call such set the rejection set $\mathcal{RS}$.

Intuitively, the interest to a user of a class c with respect to a filter set $\mathcal{FS}$ should take into account both the absolute importance of c (as explained in the previous section) and a closeness measure of c with regard to the classes in $\mathcal{FS}$. For this reason, we define:

$$\Phi(c, \mathcal{FS}) = \alpha \times \Psi(c) + (1 - \alpha) \times \Omega(c, \mathcal{FS}) \qquad (1)$$

where $\Phi(c, \mathcal{FS})$ is the interest of class c with respect to $\mathcal{FS}$, $\Psi(c)$ the absolute importance of class c, and $\Omega(c, FS)$ is the closeness of class c with respect to $\mathcal{FS}$.

Note that α is a balancing parameter in the range [0,1] to set the preference between closeness and importance for

the retrieved knowledge. An $\alpha > 0.5$ benefits importance against closeness while an $\alpha < 0.5$ does the opposite. The default α value is set to 0.5 and can be modified by the user.

There may be several ways to compute the closeness $\Omega(c, \mathcal{FS})$ of class c with respect to the classes of $\mathcal{FS}$. Intuitively, the closeness of class c should be directly related to the inverse of the distance of c to the filter set $\mathcal{FS}$. For this reason, we define:

$$\Omega(c, \mathcal{FS}) = \frac{|\mathcal{FS}|}{\sum_{c' \in \mathcal{FS}} d(c, c')} \quad (2)$$

where $|\mathcal{FS}|$ is the number of classes of $\mathcal{FS}$ and $d(c, c')$ is the minimum distance between a class c and a class c' belonging to the filter set $\mathcal{FS}$. Intuitively, those classes that are closer to more classes of $\mathcal{FS}$ will have a greater closeness $\Omega(c, \mathcal{FS})$.

We assume that a pair of classes c, c' are directly connected to each other if there is a direct association (or redefinition of association) between them or if one class is a direct subclass of the other. For these cases, $d(c, c') = 1$. Otherwise, when c, c' are not directly connected, $d(c, c')$ is defined as the length of the shortest path between them traversing associations and/or ascending/descending through class hierarchies.

As an example, Table II shows the top-10 classes with a greater value of interest when the user defines $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$ and $\alpha = 0.5$.

Results in Table II indicate that included within the top-10 there are classes that are directly connected to all members of the filter set $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$ as in the case of *SubjectOfActAppointment* ($\Omega($*SubjectOfActAppointment*$, \mathcal{FS}) = 1.0$) but also classes that are not directly connected to any class of *FS* (although they are closer).

V. Filtering HL7 Information Models

We have developed a method for filtering large models, and we have used the HL7 models as a case study for developing and experimenting with the method, and its associated tool. The method consists of four consecutive steps. The characteristics of each step are detailed below. Figure 2 presents an overview of the method and steps.

Intuitively, from a small subset of classes selected by the user the method automatically obtains a filtered information model with knowledge of interest.

Step 1: Setting the User Preferences

The first step of the method consists of prepare the required information to filter the HL7 information models according to the user preferences. Basically, the user focus on a set of classes (filter set) she is interested in and our method surrounds them with additional knowledge from the HL7 models. Therefore, it is mandatory for the user to select

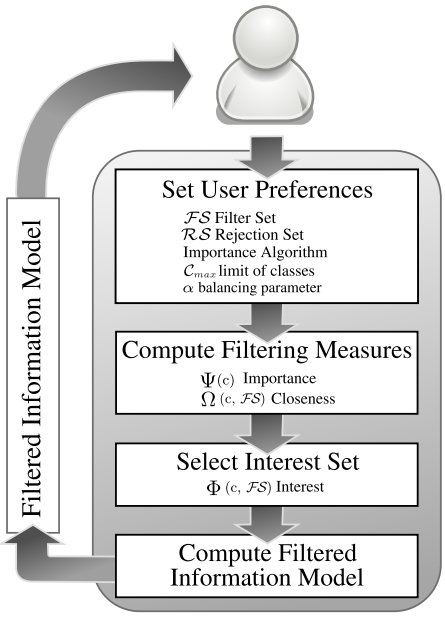

Figure 2. Method Overview.

a non-empty initial filter set $\mathcal{FS}$. An example of filter set to obtain knowledge about patient and appointments in the HL7 can be $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$.

In the same way, the user can specify a rejection set $\mathcal{RS}$ (may be empty) with those classes that have no interest to her.

In addition to the filter set, the user can decide the amount of knowledge she wants to obtain by indicating the number of additional classes ($\mathcal{C}_{max}$) the method has to select and include in the filtered information model.

Apart from that, the user has the possibility to select which importance method (see Section III) wants to be used in the following step. Also, she can include her preferences about closeness and importance by setting a value for the balancing parameter α (see (1) in Section IV).

Note that $\mathcal{RS}$, $\mathcal{C}_{max}$, the importance method, and the parameter α have default values ($\mathcal{RS} = \emptyset$, $\mathcal{C}_{max} = 10$, the default importance method is CEntityRank [15] and $\alpha = 0.5$) and therefore are all optional.

The user interaction is required only in this initial step.

Step 2: Compute Filtering Measures

The second step of the method consists in computing the metrics of importance (Ψ) and closeness (Ω) for the HL7 classes.

By definition, the importance $\Psi(c)$ of a class c is an absolute metric that depends on the knowledge represented on the whole set of HL7 models. The filtering method computes the importance of each class in the HL7 models, but this computation must be done only once. The results are valid until the HL7 models change. In our current prototype, the time required for this computation is about 2 seconds.

Table II
MOST INTERESTING CLASSES WITH REGARD TO $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$.

Rank	Class (c)	$\Psi(c)$	$d(c,$ *Patient*$)$	$d(c,$ *ActAppointment*$)$	$\Omega(c, \mathcal{FS})$	$\Phi(c, \mathcal{FS})$
1	SubjectOfActAppointment	0.11	1	1	1.0	0.7003
2	Organization	1.72	1	3	0.5	0.3552
3	Person	1.22	1	3	0.5	0.3537
4	ServiceDeliveryLocation	0.79	2	2	0.5	0.3524
5	AssignedPerson	0.72	2	2	0.5	0.3522
6	ManufacturedDevice	0.55	2	2	0.5	0.3517
7	LocationOfActAppointment	0.26	3	1	0.5	0.3508
8	ReusableDeviceOfActAppointment	0.19	3	1	0.5	0.3506
9	SubjectOfAccountEvent	0.13	1	3	0.5	0.3504
10	AuthorOfActAppointment	0.12	3	1	0.5	0.3503

On the other hand, to compute the closeness $\Omega(c, \mathcal{FS})$ of an HL7 class with regard to the filter set $\mathcal{FS}$ it is required to know the minimum distances between classes in the HL7 models (see (2) in Section IV). However, it is only necessary to compute the distance from each class in the filter set to any class out of $\mathcal{FS}$, which requires a lower computational cost. Note that the method computes the closeness only for those classes that are out of the filter set.

Step 3: Select Interest Set

The third step of the method consists in computing the interest (Φ) for each class out of the $\mathcal{FS}$. As previously shown in (1) of Section IV, the interest $\Phi(c, \mathcal{FS})$ of a candidate class c to be included in the output model is a linear combination of the importance $\Psi(c)$ and the closeness $\Omega(c, \mathcal{FS})$ taking into account the balancing parameter α.

Note that if a non-empty rejection set $\mathcal{RS}$ was defined in the first step of our method, those classes included in such set will not be considered for the final result nor their interest Φ will be computed.

The interest Φ produces a sorted ranking of HL7 classes and the method selects the top classes of that ranking until reaching the selected limit $\mathcal{C}_{max}$ specified in the first step. We call such set of classes the Interest Set. Second column of Table II shows the classes that belong to the Interest Set according to $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$ when $\mathcal{C}_{max} = 10$.

In case of two or more classes get the same interest our method is non-deterministic: it might select any of those. Some enhancements can be done to try to avoid selecting classes in a random manner, like prioritizing the classes with a higher value of closeness or importance (or any other measure) in case of ties.

Step 4: Compute Filtered Information Model

Finally, the last step of the method obtains the Interest Set of classes from the previous step and puts it together with the classes of the filter set $\mathcal{FS}$ in order to create a filtered information model with the classes of both sets.

The main goal of this step consists in filtering information from the whole HL7 information models involving classes in the filtered model. To achieve this goal, the method explores the associations, redefinitions of associations, and generalization/specialization relationships in the HL7 information models that are defined between those classes and includes them in the filtered model to obtain a connected model. The filtered information model for $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$ and the previous Interest Set is shown in Figure 3.

Our method also takes into account associations that are specified between superclasses of classes included in the filtered information model, and brings them down to connect such subclasses. An example of that behaviour is the association between *Participation* and *ActAppointment* in Figure 3. Such association is originally defined between *Participation* and *Act* (see Figure 1). Given that, *ActAppointment* is a subclass of *Act*. Such association is descended to the context of *ActAppointment* to indicate that there exists the connection with *Participation* although *Act* was not included in the Interest Set.

When descending an association there exist the case that such association could be repeated. Figure 3 shows the association between *Participation* and *ActAppointment*. Note that *Participation* is not a member of the Interest Set (see Table II). However, *Participation* has been included in the filtered information model as an auxiliary class (marked in Figure 3 with a light grey color). The rationale is that such association should be descended between each of the five subclasses (*SubjectOfActAppointment, AuthorOfActAppointment, ReusableDeviceOfActAppointment, LocationOfActAppointment and SubjectOfAccountEvent*) of *Participation* present in the Interest Set and *ActAppointment* which is not an UML compliant situation.

To avoid repeated associations our method finds the lowest common parent (LCP) for the previous subclasses, which in this case is *Participation*, includes it in the filtered information model as an auxiliary class, and descends the

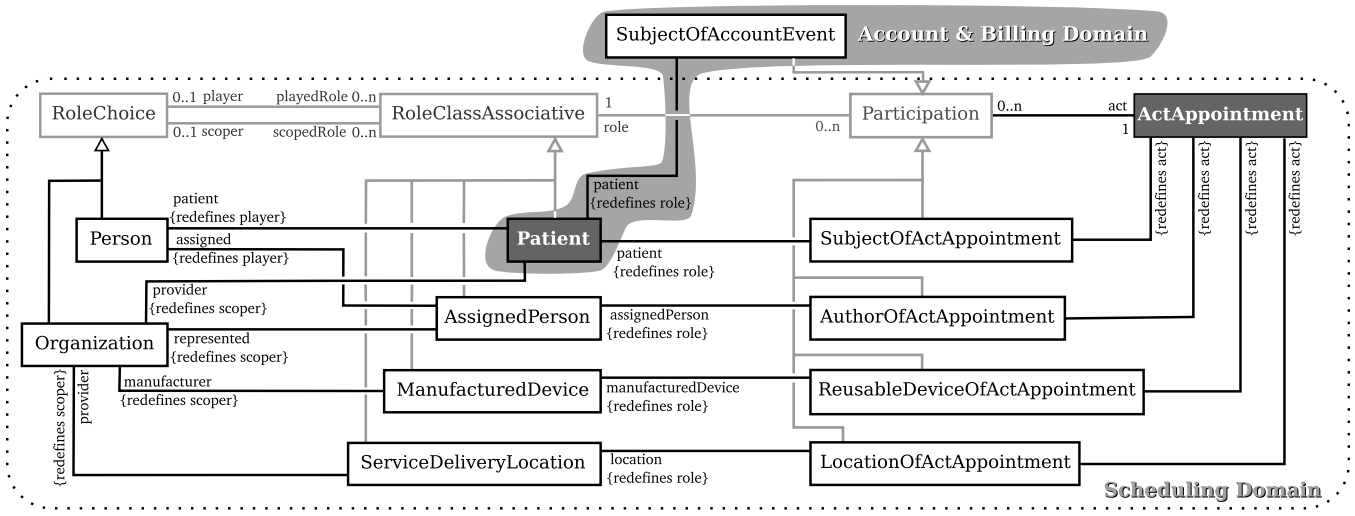

Figure 3. Filtered Information Model for $\mathcal{FS} = \{$*Patient, ActAppointment*$\}$.

association to such LCP class. The same situation occurs for *RoleClassAssociative* and *RoleChoice*, which are LCP classes included as auxiliary in the filtered information model of Figure 3.

Besides, if there are two classes in the filtered information model such that one is an indirect subclass of the other in the HL7 models, our method creates an *IsA* relationship between them in the filtered information model (marked as indirect) to indicate such knowledge. Figure 3 shows that the five subclasses of *Participation* and the four ones of *RoleClassAssociative* are indirect subclasses by marking those *IsA* relationships in a light gray color. For the case of *RoleChoice*, its subclasses are directly connected to it by means of *IsA* relationships (marked with ordinary black color).

Finally, the filtered information model presented in Figure 3 shows information about two HL7 domains: the Scheduling domain and the Account and Billing domain. By using our filtering method, a user that wanted to know about patients and appointments discovers that patients are also related to account events. This way, the user easily can compose another filter set like $\mathcal{FS} = \{$*Patient, SubjectOfAccountEvent*$\}$ to get more knowledge about them in a new iteration of our method.

VI. Evaluation

Our filtering method and prototype tool provide support to the task of extracting knowledge from the HL7 models, which has normally been done manually or with little support.

Finding a measure that reflects the ability of our method to satisfy the user is a complicated task. However, there exists related work [16], [17] about some measurable quantities in the field of information retrieval that can be applied to our context:

- The ability of the method to withhold non-relevant knowledge (*precision*)
- The interval between the request being made and the answer being given (*time*)

Precision Analysis

A correct method must retrieve the relevant knowledge according to the user preferences. The precision of a method is defined as the percentage of relevant knowledge presented to the user.

In our context, we use the concept of precision applied to HL7 universal domains (specified with D-MIM's). Each domain contains a main class which is the central point of knowledge to the users interested in such domain. The other classes presented in the domain conform the relevant knowledge related to the main class.

HL7 professionals interested in a particular domain decide about the knowledge to incorporate in it through ballots. Thus, a common situation for a user is to focus on the main class of a domain and to navigate through the D-MIM to understand its related knowledge.

To know the precision of our method, we simulate the generation of a D-MIM from its main class. We define a single-class filter set with such class and set $\mathcal{C}_{max}$ with the size of the domain. This way, we will obtain a filtered information model with the same number of classes as such domain.

In one iteration of our method, we obtain two groups of classes within the resulting filtered information model: the relevant classes to the user, that is, the ones that were originally defined in the D-MIM by experts, and the non-relevant ones. The precision of the result is defined as the fraction of the relevant classes over the total $\mathcal{C}_{max}$.

To refine the obtained result, the non-relevant classes are included in a rejection set $\mathcal{RS}$ and the method is executed

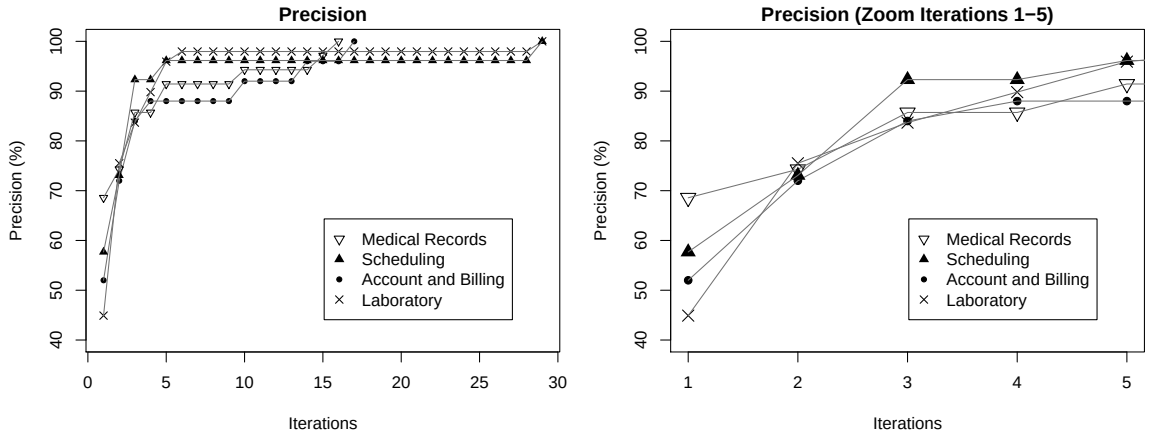

Figure 4. Precision analysis for HL7 domains.

again taking into account $\mathcal{RS}$. It is expected that the filtered information result of this step will have a greater precision.

This manner, at each iteration non-relevant classes to the user are rejected, and we know that in a finite number of steps our filtering method will obtain all the classes of the original domain. The smaller the number of required iterations until getting such domain, the better the method.

Figure 4 shows the number of iterations needed to reach the maximum precision for four of the HL7 domains. Note that right side of Figure 4 zooms in the first five iterations. The test reveals that to reach more than 80% of the relevant classes of a domain, only three iterations are required.

Time Analysis

It is clear that a good method does not only require precision, but it also needs to present the results in an acceptable time according to the user.

To find the time spent by our method it is only necessary to record the time lapse between the request of knowledge, i.e. once a filter set $\mathcal{FS}$ has been indicated by the user, and the receipt of the filtered information model.

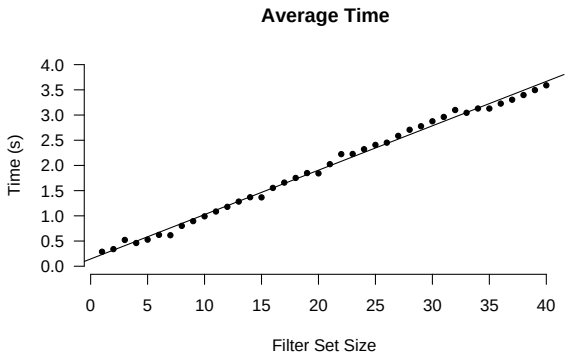

Figure 5. Time analysis for different sizes of $\mathcal{FS}$.

It is expected that as we increase the size of the filter set, the time will increase linearly. Our method computes the distances from each class in the filter set to all the rest of classes. This computation requires the same time (in average) for each class in the filter set. Therefore, the more classes we have in a filter set, the more the time our method spends in computing distances.

In our experimentation, we set our prototype tool to apply the filtering method several times with an increasing number of classes in the filter set. The average results for sizes from a single-class filter set up to a 40-classes filter set are presented in Figure 5.

According to the expected use of our method, having a filter set $\mathcal{FS}$ of 40 classes is not a common situation (although possible). Sizes of filter sets up to 10 classes are more realistic, in which case the average time does not exceed one second.

VII. Conclusions

HL7 information models are very large. The wealth of knowledge they contain makes them very useful to their potential target audience. However, the size and the organization of these models makes it difficult to manually extract knowledge from them. This task is basic for the improvement of services provided by HL7 affiliates, vendors and other organizations that use those models for the development of health systems.

What is needed is a tool that makes HL7 models more usable for that task. We have presented a method that makes it easier to automatically extract knowledge from the HL7 models. Input to our method is the set of classes the user is interested in. The method computes the interest of each class with respect to that set as a combination of its importance and closeness. Finally the method selects the most interesting classes from that models, including their defined knowledge in the original models (e.g. associations,

redefinition of associations, *IsA* relationships).

The experiments we have done clearly show that the proposed method and its associated tool provides an easier way to extract knowledge from the models. Concretely, our prototype tool recovers more than 80% of the knowledge of a D-MIM in three iterations, with an average time per iteration that for common uses does not exceed one second.

We plan to continue our work along three directions. The first is to include all HL7 models into our tool to give full support to all HL7 communities. Currently we have four D-MIMs. Experimentation with the full set of models will allow us to improve the method.

We also plan to experiment with the latest definition and nomenclature of HL7 models published by the HL7 iternational. Basically, it specifies a new level on top of the RIM model that consists on a domain analysis model (DAM) to describe business process and use cases, and a localized information model (LIM) in the bottom of the model types to adapt the R-MIMs to locale-specific requirements for structure and terminology. To take into account these two new models is a challenge that will improve our work.

Finally, another research area to explore consists in generating traceability links from the elements in the filtered model to the original models, so that it is easy to find out the origin of each element. Keeping such backward links improves the integration of different models in an interoperability context. Also, our method and tool implementing traceability could be used as an aid in the design of implementation guides for HL7 interoperability artifacts (HL7 V3 messaging and CDA R2 documents).

ACKNOWLEDGMENT

The authors want to thank the collaboration of Diego Kaminker, HL7 Education WG co-chair and HL7 International Mentoring Committee co-chair, Carles Gallego, current Chair of HL7 Spain, and Dr. Joan Guanyabens, former Chair of HL7 Spain.

We would also like to thank the people of the GMC group for their useful comments to previous drafts of this paper. This work has been partly supported by the *Ministerio de Ciencia y Tecnología* under the project TIN2008-00444, *Grupo Consolidado*.

REFERENCES

[1] Health Level Seven International, "HL7 web," feb 2010. [Online]. Available: http://www.hl7.org

[2] R. Dolin, L. Alschuler, C. Beebe, P. Biron, S. Boyer, D. Essin, E. Kimber, T. Lincoln, and J. Mattison, "The HL7 clinical document architecture," *Journal of the American Medical Informatics Association*, vol. 8, no. 6, pp. 552–569, 2001.

[3] R. Dolin, L. Alschuler, S. Boyer, C. Beebe, F. Behlen, P. Biron, and A. Shabo, "HL7 clinical document architecture, release 2," *Journal of the American Medical Informatics Association*, vol. 13, no. 1, pp. 30–39, 2006.

[4] J. Conesa, V. C. Storey, and V. Sugumaran, "Usability of upper level ontologies: The case of researchcyc," *Data & Knowledge Engineering*, vol. 69, no. 4, pp. 343–356, 2010.

[5] A. Danko, R. Kennedy, R. Haskell, I. Androwich, P. Button, C. Correia, S. Grobe, M. Harris, S. Matney, and D. Russler, "Modeling nursing interventions in the act class of HL7 RIM Version 3," *Journal of biomedical informatics*, vol. 36, no. 4-5, pp. 294–303, 2003.

[6] J. Lyman, S. Pelletier, K. Scully, J. Boyd, J. Dalton, S. Tropello, and C. Egyhazy, "Applying the HL7 reference information model to a clinical data warehouse," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, 2003, pp. 4249–4255.

[7] OMG, *Unified Modeling Language: Superstructure, version 2.1.1*, Object Modeling Group, February 2007.

[8] OMG, *Object Constraint Language, version 2.0*, Object Modeling Group, May 2006.

[9] S. Castano, V. De Antonellis, M. G. Fugini, and B. Pernici, "Conceptual schema analysis: techniques and applications," *ACM Transactions on Database Systems*, vol. 23, no. 3, pp. 286–333, 1998.

[10] D. L. Moody and A. Flitman, "A methodology for clustering entity relationship models - a human information processing approach," in *Conceptual Modeling - ER 1999, 18th International Conference on Conceptual Modeling*, ser. Lecture Notes in Computer Science, vol. 1728. Springer, 1999, pp. 114–130.

[11] Y. Tzitzikas, D. Kotzinos, and Y. Theoharis, "On Ranking RDF Schema Elements (and its Application in Visualization)," *Journal of Universal Computer Science*, vol. 13, no. 12, pp. 1854–1880, 2007.

[12] Y. Tzitzikas and J.-L. Hainaut, "How to tame a very large er diagram (using link analysis and force-directed drawing algorithms)," in *Conceptual Modeling - ER 2005, 24th International Conference on Conceptual Modeling*, ser. Lecture Notes in Computer Science, vol. 3716. Springer, 2005, pp. 144–159.

[13] C. Yu and H. V. Jagadish, "Schema summarization," in *VLDB 2006, 32nd International Conference on Very Large Data Bases*, 2006, pp. 319–330.

[14] X. Yang, C. M. Procopiuc, and D. Srivastava, "Summarizing relational databases," in *VLDB 2009, 35th International Conference on Very Large Data Bases*, 2009, pp. 634–645.

[15] A. Villegas and A. Olivé, "On computing the importance of entity types in large conceptual schemas," in *Advances in Conceptual Modeling - Challenging Perspectives, ER 2009 Workshops*, ser. Lecture Notes in Computer Science, vol. 5833. Springer, 2009, pp. 22–32.

[16] R. Baeza-Yates and B. Ribeiro-Nieto, *Modern Information Retrieval*. Addison Wesley, 1999.

[17] C. Van Rijsbergen, "Information Retrieval," *Cataloging & Classification Quarterly*, vol. 22, no. 3, 1996. [Online]. Available: http://www.dcs.gla.ac.uk/Keith/Preface.html