Customized clinical domain ontology extraction for Knowledge Authoring Tool

Taqdir Ali
UC Lab, Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Korea
taqdir.ali@oslab.khu.ac.kr

Maqbool Hussain
UC Lab, Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Korea
maqbool.hussain@oslab.khu.ac.kr

Wajahat Ali Khan
UC Lab, Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Korea
wajahat.alikhan@oslab.khu.ac.kr

Muhammad Afzal
UC Lab, Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Korea
muhammad.afzal@oslab.khu.ac .kr

Sungyoung Lee
UC Lab, Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, 446-701, Korea
sylee@oslab.khu.ac.kr

ABSTRACT
Clinical Decision Support Systems (CDSS) require a shareable and adaptable knowledge base. However, sharing and reusing the expert's knowledge is a challenging task. The proposed approach designs a web based application that acquires and adapts the clinical expert's knowledge into shareable knowledge base. The system, Intelligent Knowledge Authoring Tool (I-KAT) creates rules in the form of Medical Logic Module (MLM) using HL7 standard Arden Syntax. These rules are easily shareable with HL7 complaint clinical institutions and organizations. To achieve interoperability using MLM, the system uses a mechanism for integration of terminology standard (SNOMED CT) concepts with CDSS standard (Virtual Medical Record (vMR)). The SNOMED CT ontology is comprehensive; containing more than 0.3 million concepts but 10 – 15% concepts of total ontology is normally used in rule creation for a specific domain. Semantically defining relationships between SNOMED CT concepts and vMR concepts require domain ontology development from the SNOMED ontology. In this paper we focus on automatic extraction of domain ontology from overall SNOMED CT ontology on the basis of vMR schema concepts and their attributes mapping with corresponding SNOMED CT concepts. The extracted domain ontology will increase the efficiency and effectiveness of searching mechanism in contextual selection process.

Categories and Subject Descriptors
D.2.12 [Software Engineering]: Interoperability

H.5 [INFORMATION INTERFACES AND PRESENTATION]: Standardization

I.2.4 [Knowledge Representation Formalisms and Methods]

General Terms
Algorithms, Measurement, Performance, Design, Standardization

Keywords
Ontology extraction, Domain ontology, Authoring Tool, Decision Support System, SNOMED CT, Virtual Medical Record.

IMCOM (ICUIMC)'14, January 9–11, 2014, Siem Reap, Cambodia.

1. INTRODUCTION
Clinical Decision Support System (CDSS) is progressing domain that assists physicians in different areas of diagnosis, treatment and interventions of a patient [1]. CDSS provides recommendations and generates guidelines using expert's knowledge stored in the knowledge base. Evolutionary knowledge base thus becomes an important feature that results in extendable and maintainable knowledge base. The proposed approach contributes towards the important features of CDSS that is shareable knowledge base and integration with other systems [2]. However, standard representation of knowledge, information model and interoperability are some challenges in sharing the knowledge bases. HL7 Arden Syntax is knowledge representation standard which transform rules into shareable logic unit called MLMs [5]. To unify interfaces and allow healthcare systems to easily communicate with CDSS, HL7 vMR (Virtual Medical Record) is providing data models for CDSS input and output. HL7 vMR is derived from RIM (Reference Information Model) with the main objective of semantic interoperability [11]. Terminology standards such as SNOMED CT, LOINC are used to handle the clinical concepts of a particular domain. SNOMED CT is an international standard ontology has multilingual supports and contains more than 0.3 million active concepts [12]. To

achieve maximum degree of semantic interoperability, the SNOMED CT ontology is one of the choices to use in standard MLMs. Therefore, the use of standard information model HL7 vMR and standard terminologies such as SNOMED CT ontology in MLM provide interoperability and exchangeability features in knowledge base.

Knowledge Authoring Tool [2] transforms the physician's knowledge into standard MLM using customized and user friendly interfaces. These MLM represents the medical rules using standard information model HL7 vMR and standard terminologies SNOMED CT. The system provides user friendliness through intelli-sence facility to physicians to encode the rules. This intelli-sence depends on the vMR Schema concepts by mapping them with corresponding SNOMED CT concepts. It is based on mapping vMR concept with set of SNOMED CT concepts for the physicians to choose the most appropriate concept. The problem with large scale ontologies is the efficiency, for example SNOMED CT covers the whole healthcare domain with multiple diseases' concepts. Searching appropriate concept from such large scale ontologies affects the overall performance of the system. Domain specific applications only require subset of whole concepts thus domain specific ontology is needed [10], it will enhance the process of efficient and effective search.

Our main focus in this paper is automatic extraction of customized domain ontology from the overall large scale SNOMED CT ontology. We extract the domain ontology concepts according to vMR schema concepts and their attributes. The approach is based on seed query concept to extract specific domain ontology. Prior to domain ontology extraction, the *Ontology Importer* transforms the external representation of SNOMED CT ontology to internal representation of the system that is the ontology standard OWL format, OWL is semantic web language designed to represent rich and complex knowledge with high expressive power [13]. Maintenance of the complete semantics of concepts requires incorporating seed query concept. We use two types of seed queries, one is the *ParentSeedQuery* and second is the *ChildSeedQuery*. *ParentSeedQuery* generates the query of generalized terms as top level concepts of SNOMED CT while the *ChildSeedQuery* generates the query of domain specific terms. The system extracts all the parent concepts of terms defined by *ChildSeedQuery* up to the top level concepts defined by *ParentSeedQuery*. Both seed queries are generated by *Query Manager* module of the system. The seed queries realize the customization in domain ontology extraction. *The Query Manager* matches the vMR schema concepts and their attributes to the corresponding SNOMED CT top level concepts by using the *SNOMED vMR Mapper* sub-module.

The system works on two basic algorithms; *Upward Traversal*, and *Downward and Horizontal Traversal* Algorithms. The *Upward Traversal* algorithm extracts all the supper classes in between child seed concept and corresponding parent seed concept, while the Child concepts of these supper classes are omitted. Similarly the *Downward Traversal* part of algorithm extracts all the child concepts of defined child seed concept. The *horizontal Traversal* part of algorithm extracts only those concepts which 1) link to the child concepts of the defined child seed concept with attribute relationship and 2) the top most parent exists in parent seed concepts, otherwise the system also ignores the attribute related concepts.

Rest of the paper is structured as: Section 2 explains background of I-KAT. Section 3 discusses some related work of the domain ontology extraction and ontology segmentation. Section 4 explains the architecture, methodology and algorithms of the proposed system. Section 5 is provided with discussion on system and Section 6 concludes the paper.

2. I-KAT

To acquire shareable and interoperable expert's knowledge for Smart CDSS [3][4] we have proposed and designed a Knowledge Authoring Tool [2]. An Intelligent Knowledge Authoring Tool (I-KAT) contains three main components *Rule Editor*, *Rule Validator* and *Compilation Module* as shown in Figure 1. The *Rule Editor* provides a user friendly interface to physicians for transforming their knowledge to Smart CDSS's knowledge base. To achieve the shareability of knowledge, we use HL7 standard Medical Logic Modules (MLMs) to represent the knowledge rules. Arden syntax is well-known and established way to represent MLMs [5]. Creating and publishing rules into integrated environment I-KAT performs two tasks; 1) Transform the clinician knowledge into sharable MLMs which is stored in MLM File Repository. 2) To publish these rules into real production environment, the MLMs are compiled into clinical knowledge base to work with integrated workflows of healthcare system. To achieve interoperability and provide ease of use for physicians in writing rules, the *Rule Editor* provides the facility of automatic and intelligent fetching of the SNOMED CT concepts according to vMR schema's selected class. These processes make the system more *Interactive*, *Interoperable* and *Intelligent*.

SNOMED CT contains more than 0.3 million concepts [12], therefore searching and fetching the concepts from overall SNOMED CT ontology on base of vMR schema class will be very slow and deliberative task. In [6], the system extracts breast cancer domain relevant concepts that includes 1% of overall SNOMED CT ontology; showing 99% less concepts than overall ontology. However, each domain ontology contains 10 - 15% key concepts as well as relevant concepts of overall SNOMED CT concepts. Therefore, searching and fetching concepts from small ontologies is faster than large ontologies. I-KAT provides the intelli-sence facility from domain ontology instead of overall SNOMED CT ontology. Extraction of domain ontology is very tedious and time consuming task for ontology engineers. I-KAT overcomes this problem by providing automated extraction of domain ontology from SNOMED CT ontology.

3. RELATED WORK

There are many research groups working on domain ontology extraction. Milian K et al. [6] proposed a system to extract domain ontology; they have focused on breast cancer domain. This system extracts the domain ontology using two methods. Firstly, system extracts the key concepts of domain using seed query method. Secondly, they manually extract the concepts from existing guidelines provided by Dutch Institute for Healthcare Improvement. By using the first method, the system extracts only key concepts while the relevant concepts are omitted. The relevant concepts are extracted from guidelines and then mapped to SNOMED CT concepts manually. Manual extraction and mapping of concepts is time consuming and tedious task. The provided guidelines only cover the treatment; therefore the system

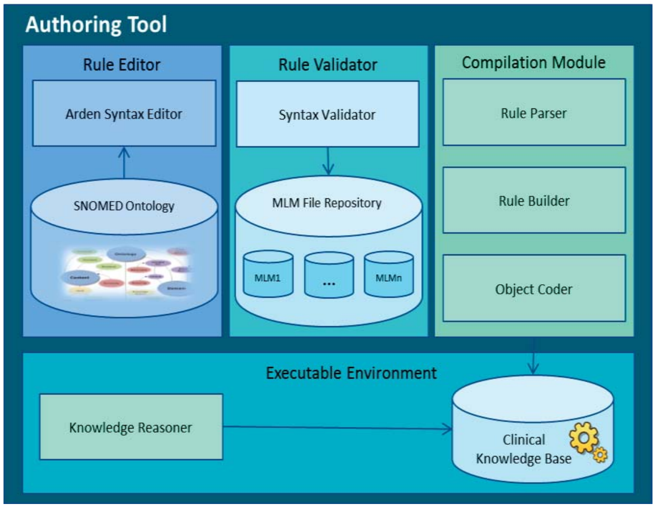

Figure 1: architecture of authoring tool for Smart CDSS [2]

only extracts the treatment related concepts while the diagnosis concepts are missing. Finally, the extracted disease-centric ontology contains only the concepts related to treatment of breast cancer disease. So this disease-centric ontology cannot cover the complete domain.

J. Allones et al. in [7] proposed a system to support semantic search across archetypes. The system provides an automated method based on SNOMED CT modularization to transform the clinical archetypes to SNOMED CT extracts. System is using different segmentation algorithms to automatically generate initial set of seed concepts from archetypes and then enhance the quality of extracted segments. The system performs graph traversal process on SNOMED CT ontology twice, once for mapping the seed concepts of archetype to SNOMED CT concepts and secondly, to extract the expected segment based on seed concepts. The system is using a virtuous approach of automated seed concepts to extract ontology segments but the scope of the extracted ontology is narrow to segments and the extracted segments may not cover the complete domain.

M. Bhatt et al. [8] introduced a system named, MOVE (Materialized Ontology View Extraction) to extract a sub-ontology from the huge ontologies. MOVE system is using different optimization schemes to maintain the semantic completeness, well-formedness and to derive sub-ontology that is highly qualitative. MOVE system contains a user defined ontology labeling component that facilitates the manipulation during the extraction process. The labeling component allows the users to provide subjective information for included/excluded concepts and attributes in the targeted domain ontology. This is a standard way that the different components of the extraction process communicate with each other but it is a tedious task for users to identify both the selected and unselected concepts and relationships. This system may not handle those concepts that are not included in both sets of selected and unselected concepts but exists in the source ontology.

J. Seidenberg et al. [9] have provided a solution to extract customized segments to solve scaling problem of big ontologies. In this system the authors have proposed some algorithms that start the extraction process with one or more classes of user's choice and create an extract or segment around those related concepts. The system handles the supper class's hierarchy; sub classes hierarchy with the help of upward traversal and downward traversal respectively; while the sibling hierarchy is not included in the extract. It is assumed that the sibling concepts are not relevant enough to be included by default but users can explicitly select them for inclusion, if they are of interest. This system is using boundary classes to define the depth limit; this approach is very useful to minimize the depth of targeted ontology but system must be knows about the chain of links that creates a list of classes to include in extract or segment.

Our system covers the complete domain concepts in extraction ontology by using *ParentSeedQuery* and *ChildSeedQuery*. These queries bring customization in domain ontology extraction. The *ParentSeedQuery* and *Horizontal Traversing* also help in extraction of relevant concepts. To identify the boundary, the system takes help from the mapping of vMR schema classes and SNOMED CT top level concepts.

4. METHODOLOGY

4.1 Architecture

Customized Domain Ontology extraction mechanism architecture design is shown in Figure 2. *Ontology Importer* transforms the external representation of SNOMED CT ontology to the internal representation of our system, which is standard OWL format [13]. The SNOMED CT is owned, maintained and distributed by the IHTSDO (International Health Terminology Standards Development Organization) [14]. The external format provided by IHTSDO is three separate files of *Concepts*, *Relationships* and *Descriptions* in text files and these files are related to each other using primary and foreign key concept. Therefore, the *Ontology Importer* module of our system transforms these concepts, *relationships* and *descriptions* to the internal representation of OWL format and stores it as *SNOMED Ontology* in our system. This new formatted *SNOMED Ontology* is able to extract the specific *Domain Ontology*.

The *Concept Loader* module of the system performs the core activity domain ontology extraction. The *vMR Ontology* consists of all the usable vMR classes and their attributes in ontology form. The *Concept Loader* loads all required vMR classes and their attributes and provide to the *Query Manager*. The *Query Manager* comprise of two main modules, one is the *SNOMED vMR Mapper* and other is the *Seed Query Creator*. The *SNOMED vMR Mapper* maps the vMR classes and their attributes with

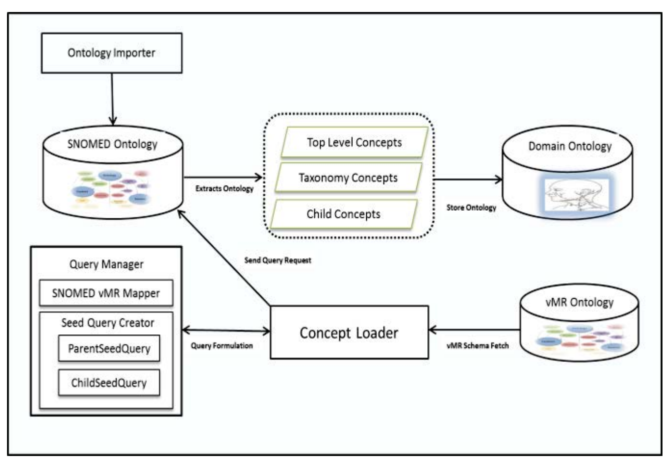

Figure 2: Architecture of domain ontology extraction

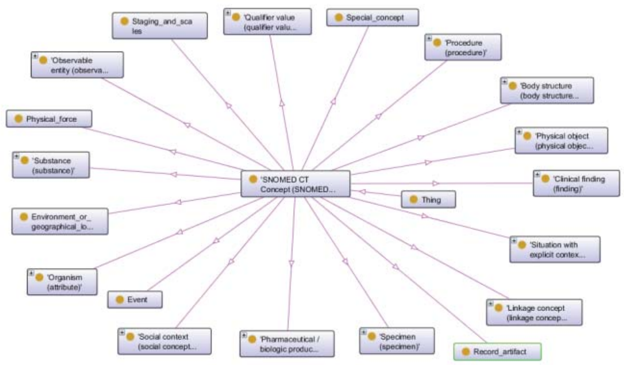

Figure 3: Top level concepts of SNOMED CT

corresponding top level concepts of SNOMED CT. The total number of top level concepts of SNOMED CT is 19 that are sub-classes of the root concept "SNOMED CT Concept", shown in Figure 3. All the remaining concepts are categorized according to the nature and semantics of these 19 top level concepts.

The *SNOMED vMR Mapper* maps these top level concepts to the required classes of vMR schema. The SNOMED CT concept "Clinical Finding" and "Observable Entity" are mapped with vMR class "ObservationBase" because the nature of both of these concepts of SNOMED CT is same to vMR class "ObservationBase". According to the user guide of SNOMED CT provided by IHTSDO in [15], concepts in the hierarchy of "Clinical Finding" represent the result of a clinical observation, assessment or judgment, and include both normal and abnormal clinical states. Similarly, the SNOMED CT concept "Observable Entity" represents the concepts regarding the observations and results after handling some procedures or answering some questions. On the other hand, the vMR class "ObservationBase" stores the objects of observations, therefore the SNOMED CT

Table 1: Mapping of vMR class's attributes with SNOMED concepts

VMR Class Attribute	SNOMED Top Concept	Purpose
observationFocus: CD	Clinical finding, Observable entity	This is the code that identifies the focus of the observation with as much specificity as available, or as required by a template. E.g., serum potassium level, hemoglobin A1c level, smoking status
observationMethod : CD	Procedure	The approach used to make the observation. E.g., direct measurement, indirect calculation, Enzyme-Linked Immunosorbent Assay.
targetBodySite: CD	Body structure	The body site where the observation is being made. E.g., left lung

concepts "Clinical Finding" and "Observable Entity" are logically mapped to the vMR class "ObservationBase". Another example is SNOMED CT concept "Procedure" that represents the broad variety of activities performed in the provision of healthcare. The *SNOMED vMR Mapper* maps this concept to the vMR class "ProcedureBase", which is series of steps taken on a subject to accomplish a clinical goal. Likewise the SNOMED CT concept "Body Structure" maps with vMR class "BodySite". Further the vMR classes have different attributes, which maps with different concepts of SNOMED CT, in such type of cases a single vMR class may be mapped with more than one SNOMED CT concepts. Table 1 shows the mapping of the attributes of vMR class "ObservationBase". The *SNOMED vMR Mapper* maps the attribute "observationFocus" with SNOMED concepts "Clinical Finding" and "Observable Entity" while attribute "observationMethod" maps with SNOMED concept "Procedure" and the attribute "targetBodySite" maps with "Body Structure". The *SNOMED vMR Mapper* passes the mapped combination of vMR and SNOMED CT to the *Seed Query Creator*.

The *Seed Query Creator* creates two types of queries *ParentSeedQuery* and *ChildSeedQuery* to maintain the semantics and relationships between the child concepts and parent concepts. The *ParentSeedQuery* usually contains the top level concepts of the SNOMED CT, while the *ChildSeedQuery* contains disease-centric concepts. The *ChildSeedQuery* plays a vital role in customization of ontology. In case of Head and Neck Cancer disease, the tentative *ChildSeedQuery* will include all the concepts according to Head and Neck cancer disease itself underneath "Clinical Finding" of SNOMED CT concept. Similarly all the procedures, which are related to Head and Neck cancer are covered in *ChildSeedQuery* of "Procedure". Table 2 shows the tentative *ChildSeedQuery* for "Clinical Finding", "Procedure" and "Body Structure". These seed queries passes to the *Concept Loader*, and different algorithms such as *Upward Traversal* and *Downward and Horizontal Traversal* are implemented to extract the child concepts, parent concepts and attribute related concepts from the overall *SNOMED Ontology* and finally populating the *Domain Ontology*.

4.2 Domain Ontology Extraction Algorithms

The main goal of *Concept Loader* is to extract domain ontology from the terminology standard ontology. In our case the domain

Table 2: ChildSeedQuery

VMR Class	SNOMED Concept	ChildSeedQuery
ObservationBase	Clinical Finding	Malignant tumor of head and/or neck (disorder), Head and Neck tumor, Head and Neck cancer
ProcedureBase	Procedure	Chemotherapy, Neck dissection, Reconstructive surgery, radiation therapy
targetBodySite: CD	Body structure	Lip,Oral Cavity, nasal Cavity, Pharynx, Larynx

ontology is based on Head and Neck Cancer concepts extracted from the SNOMED ontology that behaves as terminology standard ontology. We have implemented two main algorithms *Upward Traversal*, and *Downward and Horizontal Traversal* algorithm. These algorithms use *ParentSeedQuery* and *ChildSeedQuery* to extract domain ontology.

4.2.1 Upward Traversal Algorithm

The *Upward Traversal* algorithm finds the *ParentSeedQuery* and extracts the level of SNOMED CT concepts accordingly. Then algorithm finds the main *ChildSeedQuery* concept when it already exists in *Domain Ontology* with semantics then algorithm skips that concept. In contrast, either concept is missing or semantics is missing then algorithm starts the upward traversing and extracts parent, parent of parent and parent of parent of parent up to the top concept of SNOMED CT like “Clinical Finding”. But in upward traversing the algorithm skip all the Childs as well as sibling concepts of each parent concept. Suppose the *ChildSeedQuery* “Malignant tumor of head and/or neck (disorder)” has a parent concept “Malignant neoplastic disease” and it has parent concept “Neoplastic disease”, the *Upward Traversal* algorithm extracts only the parent concepts and ignores the Childs and siblings of these parents and reaches up to the “Clinical Finding” concept. Algorithm 1 explains the procedure of the *Upward Traversal* algorithm.

Algorithm 1: Upward Traversal Algorithm

```
Input: TSO, PSQ, CSQ /*where TSO is Terminology Standard Ontology, PSQ is
       Parent Seed Query, and CSQ represents Child Seed Query*/
Output: DO /*Domain Ontology*/

begin
  /*Execute Parent Seed Query*/
  PSQ.Execute(TSO)
  ListTopLevelConcepts ⟵ TSO.ExtractTopLevelConcepts()
  DOConcepts ⟵ DO.Store(ListTopLevelConcepts)
  /*Execute Child Seed Query*/
  CSQ.Execute(TSO)
  for C_i ∈ CSQ do
    TSO.Find(C_i)
    PC_i ⟵ C_i.AccessParent()
    C_i.MakeRelation(PC_i)
    while PC_i ∉ DOConcepts do
      PC_i.MakeRelation(PC_i.AccessParent())
      PC_i ⟵ PC_i.AccessParent()
      DO.Store(PC_i)
    end
  end
end
```

Terminology Standard Ontology (TSO), *ParentSeedQuery* (PSQ) and *ChildSeedQuery* (CSQ) are the inputs to the *Upward Traversal* Algorithm. First the *ParentSeedQuery* is executed using *CSQ.Execute()* method and extracts the top level concepts. These concepts are stored in *ListTopLevelConcepts* using *ExtractTopLevelConcepts*() method. Secondly, the *ChildSeedQuery* executes recursively each concepts *Ci* and searches the concept in *TSO*, the parent of *Ci* saves to *PCi* by *AccessParent*() method, and creates the is-A relationship between the *Ci* and *PCi* using *MakeRelation*() method. This process continues till the top level of concept reaches, that is also part of the *PSQ*. The output of this algorithm is *Domain Ontology* (*DO*) with only hierarchical relationships.

4.2.2 Downward and Horizontal Traversal Algorithm

The *Downward and horizontal Traversal* algorithm finds and extracts all the child concepts as well as attribute related concepts of the *ChildSeedQuery* defined concepts. For example, the *ChildSeedQuery* concept “Malignant tumor of head and/or neck (disorder)” has some child concepts; they have their own child concepts, so *Downward and horizontal Traversal* algorithm extracts all the child concepts. On the other hand, the concept “Malignant tumor of head and/or neck (disorder)” has also an attribute relationship with concept “Head and neck structure (body structure)”. The *Downward and horizontal Traversal* algorithm also checks the corresponding *ParentSeedQuery* of this attribute relationship concept, if it exists then extracts its range concept as well as all the child concepts of this particular range concept. If the corresponding *ParentSeedQuery* doesn’t exist then algorithm discards that particular attribute/range concept. Using *ChildSeedQuery* and *ParentSeedQuery* the system can easily find the boundary classes of the domain ontology. Algorithm 2 represents the working of the *Downward and Horizontal Traversal* algorithm.

Algorithm 2: Downward and Horizontal Traversal Algorithm

```
Input: TSO, CSQ /*where TSO is Terminology Standard Ontology, and CSQ
       represents Child Seed Query*/
Output: DO /*Domain Ontology*/

begin
  /*Execute Child Seed Query*/
  CSQ.Execute(TSO)
  for C_i ∈ CSQ do
    TSO.Find(C_i)
    ListChildren ⟵ C_i.GetAllChildren()
    DO.Store(ListChildren)
    RangeC_i ⟵ C_i.GetRelationship()
    PC_j ⟵ RangeC_i.AccessParent()
    RootNode ∈ TSO
    while PC_j ≠ RootNode do
      PC_j ⟵ PC_j.AccessParent()
      if PC_j ∈ DOConcepts then
        ListChildren_i ⟵ RangeC_i.GetAllChildren()
        DO.Store(ListChildren_i)
        ApplyUpwardTraversalAlgorithm(RangeC_i)
      end
      else
        RangeC_i.Discard()
      end
    end
  end
end
```

The *Downward and Horizontal Traversal* algorithm works to extract the child concepts as well as attribute relationship’s concepts. Again the whole *Terminology Standard Ontology* (*TSO*) and *ChildSeedQuery* (*CSQ*) are inputs for this algorithm. *CSQ* executes for each concept *Ci*, finds the concepts and searches for all child concepts to store in a list *ListChildren* using method *GetAllChildren*(). *GetRelationship*() method returns the range of concept *RangeCi* if that concept has attribute relationship with any other concept then algorithm finds the parent of *RangeCi* and stores in *PCj* by *AccessParent*() method. Recursively each parent node checks with *RootNode* and accesses the parent of each concept by *AccessParent*() method if it exists in *DOConcepts* then extracts all the child concepts by *GetAllChildren*() method and store in ontology by Store() method. To maintain the hierarchical relationships and attribute relationships it also apply the *Upward Traversal* algorithm using *ApplyUpwardTraversalAlgorithm*() method. On the other hand, if *PCj* concept does not exists in *DOConcepts* then *Discard*() method discards the attribute relationship concept.

5. DISCUSSION

Our proposed system extracts the domain ontology from SNOMED CT ontology on the basis of vMR specifications. One of the advantages of using vMR base extraction is to easily define the boundary classes. Many existing systems for domain ontology extraction and ontology segmentation face the limitation of defining the boundary classes. The required vMR classes and corresponding SNOMED CT top level concepts define the logical boundary to extract domain ontology, and algorithms extract the concepts within the scope of defined *ParentSeedQuery*. The *Upward Traversal* algorithm extracts parent concepts up to top level concept without Childs and siblings, while *Downward and Horizontal Traversal* algorithms extract the Child concepts as well as attribute related concepts if the corresponding top level concept is included in *ParentSeedQuery*, so there is no need to define the boundary classes explicitly. The second advantage is to provide automatic and contextual selection of concepts like intelli-sence in *Rule Editor* of I-KAT system.

Some existing systems also face some issues to extract the relevant concepts; those systems lack relevancy among the different concepts in different hierarchies. Some of those systems extract the relevant concepts manually while some define the relevant concepts as seed query. In the proposed system, this limitation is handled using mapping vMR schema classes and their attributes to the top level concepts of SNOMED CT. For example “Malignant tumor of head and/or neck (disorder)” comes under the “Clinical Finding” hierarchy and another concept “Head and neck structure”, which comes under the “Body Structure” hierarchy. Generally the Head and Neck Tumor can affect different targeted body sites like Head and Neck, Oral Cavity, Nasal cavity and many more. Therefore, some logical relationship exists between these concepts. Our system’s algorithm *Downward and Horizontal Traversal* finds the attribute relationship like “Finding Site (attribute)” of “Malignant tumor of head and/or neck (disorder)” with “Head and Neck Structure (Body Structure)”. After finding the attribute relationship, the algorithm checks the corresponding top level parent of the relationship’s range concept using *ParentSeedQuery.* If it’s parent exists then algorithm extracts that concept along with child concepts otherwise it ignores that concept. It is a reliable way to find the boundary of domain ontology at run time of extraction.

6. CONCLUSION AND FUTURE WORK

To create shareable knowledge base for CDSS, the MLM is an important standard. Interoperability of sharing knowledge base using MLM depends on healthcare standard knowledge representation and on standard terminologies usage. Therefore, our system covers interoperability by using SNOMED CT standard terminologies and knowledge representation in standard MLM. SNOMED CT is a large scale ontology covering more than 0.3 million concepts. Therefore, domain ontology in the proposed system provides the optimized way for searching concepts and using them for intelli-sence purpose. The extracted domain ontology provides the contextual selection of SNOMED CT concept facility to physicians in creating rules.

The system extracts the concepts by the seed queries. These seed queries are create by the *Seed Query Creator* using predefined concepts. Currently these concepts have defined by the expert physician of Head and Neck Cancer domain. In future we will work to automate the process of seed query creation process. It will reduce the dependency on experts.

7. ACKNOWLEDGEMENTS

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2009-(C1090-0902-0002)).

8. REFERENCES

[1] Achour, S. L., Dojat, M., Rieux, C., Bierling, P., & Lepage, E. 2001. A UMLS-based knowledge acquisition tool for rule-based clinical decision support system development. Journal of the American Medical Informatics Association. 8, 4 (2001), 351-360.

[2] Ali, T., Hussain, M., Khan, W.A., Afzal, M., & Lee, S., 2013. Authoring Tool: To acquire sharable knowledge for Smart CDSS, 35th IEEE Annual International Conference of the Engineering in Medicine and Biology (EMBC’13), July 3-7, 2013.

[3] Hussain, M., Khan, W. A., Afzal, M., & Lee, S. 2012. Smart CDSS for smart homes. In Impact Analysis of Solutions for Chronic Disease Prevention and Management pp. 266-269 (2012), Springer Berlin Heidelberg.

[4] Hussain, M., Khattak, A. M., Khan, W. A., Fatima, I., Amin, M. B., Pervez, Z., et.al & Latif, K. 2013. Cloud-based Smart CDSS for chronic diseases. Health and Technology, 1-23 (2013).

[5] Gietzelt, M., Goltz, U., Grunwald, D., Lochau, M., Marschollek, M., Song, B., & Wolf, K. H. 2012. Arden2ByteCode: a one-pass Arden Syntax compiler for service-oriented decision support systems based on the OSGi platform.Computer methods and programs in biomedicine, 106, 2 (2012), 114-125.

[6] Milian, K., Aleksovski, Z., Vdovjak, R., ten Teije, A., & van Harmelen, F. 2010. Identifying disease-centric subdomains in very large medical ontologies: a case-study on breast cancer concepts in SNOMED CT. Or: finding 2500 Out of 300.000. In Knowledge Representation for Health-Care. Data, Processes and Guidelines pp. 50-63 (2010). Springer Berlin Heidelberg.

[7] Allones, J. L. I., Taboada, M., Martinez, D., Lozano, R., & Sobrido, M. J. 2013. SNOMED CT module-driven clinical archetype management. Journal of biomedical informatics, 46, 3 (2013), 388-400.

[8] Bhatt, M., Flahive, A., Wouters, C., Rahayu, W., Taniar, D., & Dillon, T. 2004. A distributed approach to sub-ontology extraction. In Advanced Information Networking and

Applications, 2004. AINA 2004. 18th International Conference on (Vol. 1, pp. 636-641). IEEE.

[9] Seidenberg, J., & Rector, A. 2006. Web ontology segmentation: analysis, classification and use. In Proceedings of the 15th international conference on World Wide Web pp. 13-22 (May 2006). ACM.

[10] Bhatt, M., Wouters, C., Flahive, A., Rahayu, W., & Taniar, D. 2004. Semantic completeness in sub-ontology extraction using distributed methods. InComputational Science and Its Applications–ICCSA 2004 (pp. 508-517). Springer Berlin Heidelberg.

[11] Ryan, A. 2006. Towards semantic interoperability in healthcare: ontology mapping from SNOMED-CT to HL7 version 3. In Proceedings of the second Australasian workshop on Advances in ontologies-Volume 72 pp. 69-74 (December 2006). Australian Computer Society, Inc.

[12] Summary of SNOMED CT Benefits, http://www.ihtsdo.org/snomedct/whysnomedct/benefits/

[Visited on September 2013]

[13] Web Ontology Language (OWL), http://www.w3.org/2001/sw/wiki/OWL
[Visited on September 2013]

[14] SNOMED CT, http://www.w3.org/2001/sw/wiki/OWL

[Visited on September 2013]

[15] SNOMED CT – Documentation July 2013, http://ihtsdo.org/fileadmin/user_upload/doc/

[Visited on September 2013]