```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#">
<Model rdf:ID="ISO9126">
<evaluatedBy>
<rdf:Seq>
<rdf:li>
<View rdf:ID="External_Quality">
<define>
<rdf:Seq>
<rdf:li>
<Characteristic rdf:ID="Functionality">
<composedBy>
<rdf:Seq>
<rdf:li>
<Characteristic rdf:ID="Suitability">
<meausuredBy>
<Metric rdf:ID="Functional_adequacy">
<valuedBy>0.85</valuedBy>
</Metric>
```

Figure 5a : RDF file

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
 <rdfs:Class rdf:ID="Model"/>
 <rdfs:Class rdf:ID="View"/>
 <rdfs:Class rdf:ID="Characteristic"/>
 <rdfs:Class rdf:ID="Metric"/>
 <rdfs:Property rdf:ID="evaluatedBy">
              <rdfs:domain rdf:resource="#Model"/>
              <rdfs:range rdf:resource="#View"/>
 </rdfs:Property>
 <rdfs:Property rdf:ID="define">
              <rdfs:domain rdf:resource="#View"/>
              <rdfs:range rdf:resource="#Characteristic"/>
 </rdfs:Property>
 <rdfs:Property rdf:ID="composedBy">
              <rdfs:domain rdf:resource="#Characteristic"/>
              <rdfs:range rdf:resource="#Characteristic"/>
 </rdfs:Property>
 <rdfs:Property rdf:ID="meausuredBy">
             <rdfs:domain rdf:resource="#Characteristic"/>
```

Figure 5b: RDFS file

3.3.2 SPARQL

SPARQL (SPARQL Protocol for RDF and Query Language) is one of the key technologies of the Semantic Web, it is a query language and protocol -such as SQL for relational databases, which allows searching, adding, modifying or deleting RDF data. It was created by W3C and became an official recommendation 15 January 2008. Above is an example of SPARQL query that returns a list of characteristics from the knowledge base RDF (Figure 6) type:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT  * WHERE
{
?s rdf:type  http://www.univmlv.fr/~ocure/etudSchema.rdf#Characteristic
}
Resultants
-----------------------------------------------------------------------
( ?s = <file:///C:/Users/RDFData.rdf#Functionality_compliance> )
( ?s = <file:///C:/Users/RDFData.rdf#Security> )
( ?s = <file:///C:/Users/RDFData.rdf#Interoperability> )
( ?s = <file:///C:/Users/RDFData.rdf#Accuracy> )
( ?s = <file:///C:/Users/RDFData.rdf#Suitability> )
( ?s = <file:///C:/Users/RDFData.rdf#Functionality> )
```

Figure 6: The result of executing a SPARQL query on the RDF knowledge base

4. QMGenerator Framework

4.1 Principles and Functionality

The QMGenerator is Framework for the software quality supporting the SQuaM approach for the software quality, it offers an environment allowing to make the modeling, the evaluation, the improvement of the quality models and as well as the saving and the sharing of the models used and improved in a knowledge base (Figure 7). The framework QMGenerator provide the following functions:

- Quality Modeling
- Quality evaluation.
- Quality Improvement
- Quality storing and sharing in the knowledge base

4.2 Process and Architecture

4.2.1 Modeling Process

The modeling of the models of quality is done by respecting a metamodel of software quality which we have already defines in [2], the generated models respect the following the hierarchy: Views, characteristics (and subcharacteristics) and the metric (Figure 8). Considering the importance of XML technologies in the computer sciences and in particular the Web, QMGenerator instantiate quality models in a file RDF/XML. In the phase of definition of metric, QMGenerator presents to the users a glossary of metric to simplify their tasks, this glossary contains the syntax of each metric (ID, Name, Description, Reference, Formula, Audience, scaleType, Value, Interpretation, ...).

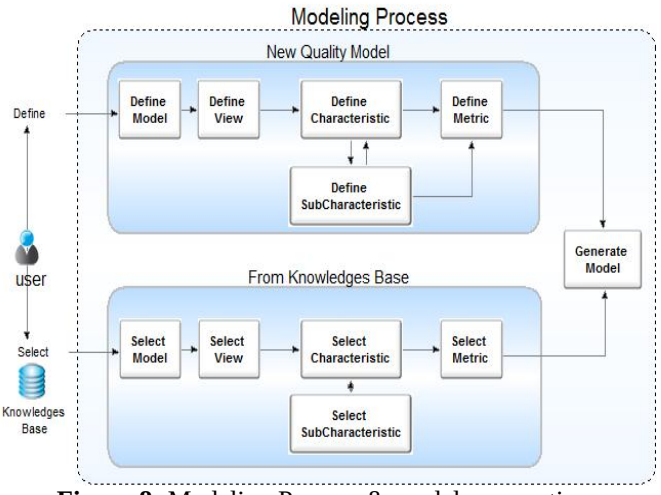

Figure 8: Modeling Process & models generation

4.3 Evaluation process

The evaluation of the quality is made through the generated quality model, developers (or other participants responsible for the quality) associate values to the metrics after measuring them. And then, the QMGenerator, starting from these measured values during the life cycle of software, creates a digital or graphic representation of these characteristics composing the model of quality and creating a software quality report (Figure 9).

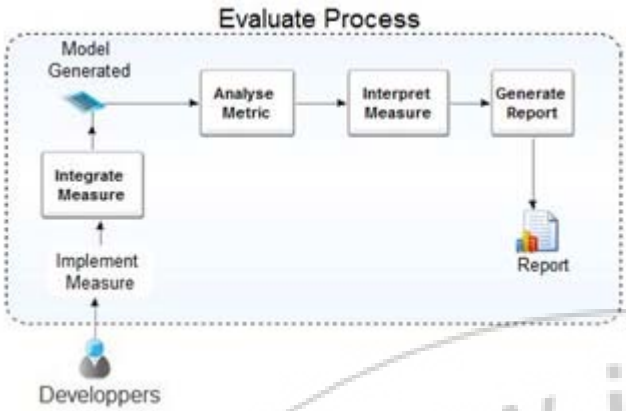

Figure 9: Evaluation Process

4.4 Improvement Process

The QMGenerator allows quality managers to make improvements in the model, these improvements include making additions, updating or deletions of views, features or metrics. Once these improvements are made, the model is a valid shareable knowledge among all members of the community of software engineering.

4.5 Technical Architecture

The QMGenerator Framework is developed by using java language in Eclipse environment , we used RDFS to the modeling and representation of the ontology domain, we used the RDF files to save knowledge in the form of triplets, and we used the JENA API for the management of knowledge (Figure 11). Jena is a Java framework for building Semantic Web applications. It provides an extensive Java libraries for helping developers develop code that handles RDF, RDFS, RDFa, OWL and SPARQL in line with published W3C recommendations. Jena includes a rule-based inference engine to perform reasoning based on OWL and RDFS ontologies, and a variety of storage strategies to store RDF triples in memory or on disk.

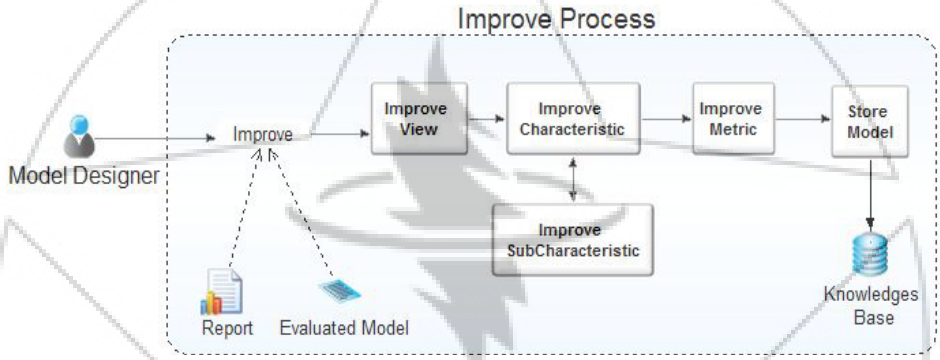

Figure 10: Improvement Process & sharing knowledge

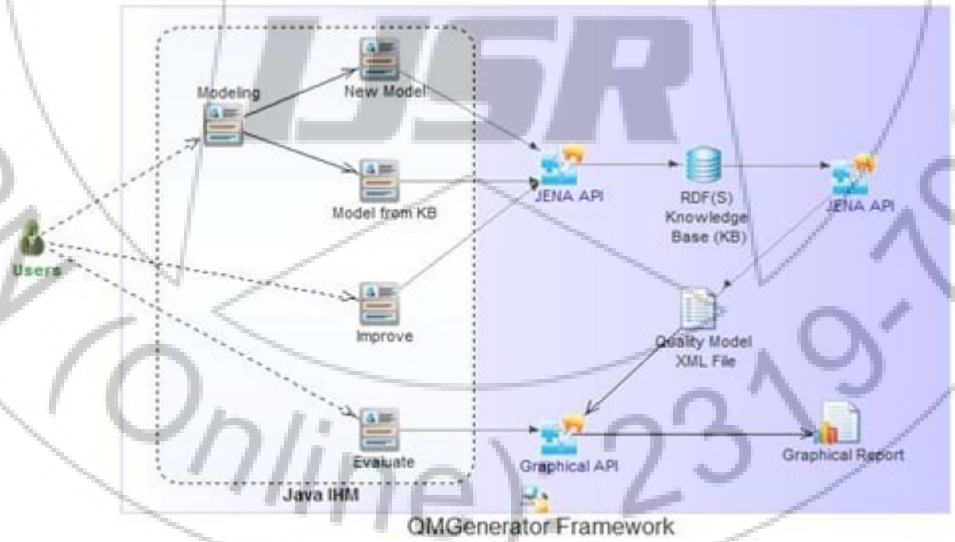

Figure 11: Technical Architecture

5. Conclusion

In this article, we presented our QMGenerator Framework for software quality based model, as well as a knowledge base of quality models to facilitate the sharing of knowledge and expertise in this area. We shall, in future work , implement our Framework in a software development project, and direct our Framework to other sectors such as industry, HR, ...

References

[1] Comparative Study of Software Quality Models, Youness BOUKOUCHI et al, the International Journal of Computer Science Issues, April 2013.

[2] A MetaModel for Quality Software Based on the MDA Approach, Youness BOUKOUCHI et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4390-4393
[3] An Approach To Software Quality Model Based (SQuaM):QMGenerator a framework supporting this approach, Youness BOUKOUCHI et al. International Journal of Science and Research (IJSR), ISSN: 2319-7064,Volume 3 Issue 6, June 2014
[4] A Quality Model for Design Patterns .Khashayar Khosravi and Yann-Gaael Gueheneuc, Summer 2004
[5] La mesure des modèles par les modèles : une approche generative, Martin Monperrus, octobre 2008.
[6] Software Engineering Metrics: What Do They Measure and How Do We Know? Cem Kaner and Walter P. Bond, 10TH INTERNATIONAL SOFTWARE METRICS SYMPOSIUM, METRICS 2004.
[7] Transition De Modèles De Connaissances Un Système De Connaissance Fondé Sur Owl, Graphes Conceptuels Et Uml, Thomas RAIMBAULT, novembre 2008, UNIVERSITÉ DE NANTES.
[8] Ontology, Tom Gruber, the Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2009
[9] Génération d'ontologies dirigée par les modèles, Guillaume Hillairet, IDM 2008, Laboratoire L3I, Université de La Rochelle
[10] Graphes RDF et leur Manipulation pour la Gestion de Connaissances, Fabien L. Gandon, novembre 2008, INRIA Sophia Antipolis
[11] Langage d'interrogation SPARQL pour R D F, Eric Prud'hommeaux et Andy Seaborne, janvier 2008, W3C.
[12] Resource Description Framework (RDF):Concepts and Abstract Syntax, W3C Recommendation 10 February 2004
[13] RDF Schema 1.1, W3C Recommendation 25 February 2014
[14] Modèles de mesure de la qualité des logiciels, Karine Mordal, Jannik Laval et Stéphane Ducasse, November 7, 2011
[15] Understanding The Model Driven Architecture (MDA), Sinan Si Alhir, Fall 2003, Volume 11 N°3 Page 18, ISSN 1023-4918
[16] THE GOAL QUESTION METRIC APPROACH, Victor R. Basili1 et al, Encyclopedia of Software Engineering, 2 Volume Set, 1994
[17] IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications
[18] ISO/IEC TR 9126, Software engineering –Product quality – Part 1,2,3. 2002-03-15.
[19] "Enterprise Information Integration and the OMG's MDA and MOF", Randall M. Hauch, OMG's Third Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise October 21-24, 2002
[20] "A Metamodel for Specifying Quality Models in Model-Driven Engineering", Parastoo Mohagheghi and VegardDehlen, Engineering Research Institute, University of Iceland , 2008
[21] Definitions and Approaches to Model Quality in Model-Based Software Development – A Review of Literature, PARASTOO MOHAGHEGHI, VEGARD DEHLEN, TOR NEPLE, Engineering Research Institute, University of Iceland , 2009
[22] 12 Steps to Useful Software Metrics, Linda Westfall,The Westfall Team, westfall@idt.net