

Development of Semantic Enabled Engineering Soil Classification Along With Visualisation of Particle Size Distribution Curve Application

Srinivas Kumar P, Raja Reddy K

Abstract: Soil classification is the basic knowledge which a geotechnical engineer needs to have before embarking on the construction of projects like highway or metro construction. With the advent of semantic web technologies it is now possible for the humans and machines to collaborate by way of understanding the underlying meaning of the soil classification concepts. An innovative approach is discussed in this paper where artificial intelligence enabled soil classification is developed along with the visualization of particle size distribution curve using R language and owl technologies

Keywords: Ontology; owl; R; rdf; semantic web.b

I. INTRODUCTION

Soil classification is one of the primary pre requisites which any geotechnical engineer need to have, either in the underground construction works or the high way construction works. The soil classification needs to be conducted periodically and at regular intervals of alignment of highway or metro construction. For ex-ample quality assurance handbook for rural roads [1] states that the soil classification tests need to be performed for each source for one km or part thereof. The sheer volume of the soil classification reports that need to generated and stored for easy management of reports and quick retrieval becomes an onerous task for huge projects such as metro construction or high way construction where the reports are prepared manually and stored in files for access and retrieval.

More over in the recent times with the advancement of the information technology, the reports can be generated by machines and they can be stored and retrieved by the machines. In that direction the work is underway [2]. Also AI techniques combined with recent advancements in internet of things , web of things, and semantic web – jointly referred to as semantic web – promise to play an important role to play in industry 4.0[3,16,17]. As a consequence a driverless excavator[4] when mounted with a semantic enabled application can understand with regard to the type of the soil it is excavating and if the soil is found to be suitable for the construction of the highway construction [5] it can load it in the truck. More over in the metro construction for TBM, the type of the cutter head is dependent on the type of soil in which the tunnel is required to be constructed. There is every possibility that TBM may encounter soils which are different from the one it was earlier designed for. As the consequence with today's technology the schedule will delay and cost of the project will go up [6]. In that case if the TBM is able to understand the type of the soil it is going to cut by way of mounting a semantic enabled application then

autonomous

blade changing feature can be incorporated to the TBM in future. In light of the above there is a need for the soil classification results to be generated and interpreted by humans and machines. Using R language the soil classification package implementation is already in place [7]. However the machines cannot understand what the result of the soil classification by using results generated by standalone R application.

Ontologies[8,9,10,11,12,13,14] pertaining to soil and soil classification were developed earlier , however the visualization of the data and predicting the values of D10,D30,D60 values from the particle size distribution from the graph is not possible using owl ontology alone.

There are semantic enabled applications[16] which were developed using languages other than R.

In this paper an attempt has been made by having an innovative approach by taking data visualization features of R and combining it with the soil classification ontology so that the machines and the humans can interpret and understand the soil classification results.

The application architecture is described, followed by the application flow.

The application was developed using shiny web server[18] , post-gresql[19] as database and rOntorion[20] as reasoner.

The code snippets which were used for the development of the application are presented.

Also the sequence of the development of the application is pre-sented.

For enhancing the usability, rhandsontable is adopted for human application interaction.

Summary and scope for the future work is presented at the end of the paper.

II. DEVELOPMENT PROCEDURE OF A WEB APPLICATION FOR ENGINEERING SOIL CLASSIFICATION USING ARTIFICIAL INTELLIGENCE

Development of Semantic Enabled Engineering Soil Classification Along With Visualisation of Particle Size Distribution Curve Application

2.1. Application architecture

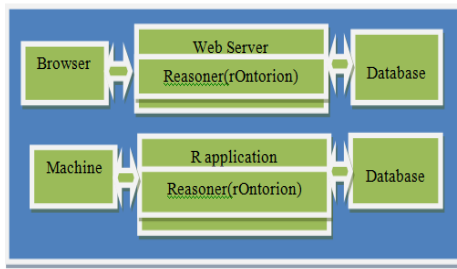


Figure 1: Application architecture

2.2. Software used for the application development

2.2.1. Software

Sr No	Software description	Version
1	Postgresql database	4
2	R	3.4
3	R studio	1.1.442
4	Shiny web server	v1.5.8.921
5	rOntorion	0.1-6

Table 1: Software's used

2.3. Application flow

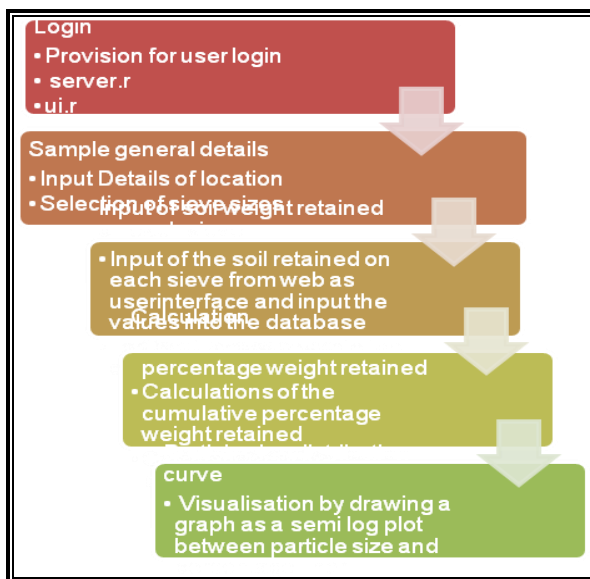


Figure 2: Application flow

2.4. Flow of web application for engineering soil classification using artificial intelligence

2.4.1. Provision for login

The sample Login code for the shiny web as follows.

In ui.R

- fluidPage(
- tagList(
- div(id = "login",
- wellPanel(textInput("username", "username"),
- passwordInput("passwd", "password"),
- actionButton("Login""Login"))))

in server.R

The code for the database connection to the postgresQL is as follows

- pw<- { "Your password" }

- con <- dbConnect(RPostgres::Postgres(),
- host='localhost',port='5432' ,
- dbname=Your DBname ,user='Your username' ,
- password=pw)
- rm(pw)

The sample login code for the server.R

- lg<-dbGetQuery(con,paste0("select * from soiluser where username='",input\$username,"'"))
- my_username<-lg\$username
- my_password<-lg\$password
- if(length(my_username)>0&&my_username==input\$username)
- {if(length(my_password)&&my_password==input\$password)
- { print("login successful....") }}

The login screen will be as below



Figure 3: Login screen

2.4.2. Sample general details

2.4.2.1. Input details of location

The sample code for the input details will be as below.

In ui.R

- mainPanel(textInput("samplename", "SampleName:"),textInput("location", "Location:"),textInput("wtsoil", "Weight of soil sample :"), actionButton("save", "submit", class = "btn-info"))

In server.R : The code for the sample input details is given below.

- values <- reactiveValues()
- values\$DT <-data.frame(Samplename=NA, location=NA, wtsoil=NA)

The input details screen is given below



Figure 4: Input details screen

2.4.2.2. Selection of sieve sizes

The selection for the sieve sizes code will be as follows.

The table created by using rhandsontable package. Sample code is as follows

In ui.R:

- fluidRow(box(rHandsontableOutput("hot3", height=200,width=100))),
- In server.R:
- xyz<-dbGetQuery(con,"select * from sievesizes table")
- rhanst1<-rhandsontable(do.call(cbind, lapply(1:1, function(i)data.table(xyz))),width=250,height=300
- output\$hot3<- renderRHandsontable({rhanst1})

The selection of sieve sizes screen is as follows.

	sieve_size_in_mm	status
42	0.425	<input checked="" type="checkbox"/>
43	0.355	<input type="checkbox"/>
44	0.3	<input checked="" type="checkbox"/>
45	0.25	<input type="checkbox"/>
46	0.212	<input type="checkbox"/>
47	0.18	<input type="checkbox"/>
48	0.15	<input checked="" type="checkbox"/>
49	0.125	<input type="checkbox"/>
50	0.106	<input type="checkbox"/>
51	0.09	<input type="checkbox"/>
52	0.075	<input checked="" type="checkbox"/>
53	0.002	<input type="checkbox"/>

Figure 5: Sieve size Details screen

2.4.3. Input of soil weight retained on each sieve

Input of the soil retained on each sieve from web as user interface and input the values into the database is as shown below

samplename	sieve_size_in_mm	wt_retained_in_each_sieve_gm
1 Raja	37.50	551.00
2 Raja	19.00	200.00
3 Raja	9.50	100.00
4 Raja	4.75	50.00
5 Raja	2.36	25.00
6 Raja	1.18	12.00
7 Raja	0.60	6.00
8 Raja	0.3	3.00
9 Raja	0.15	1.20
10 Raja	0.075	0.60

Figure 6: Display of soil weight retained data on each sieve

2.4.4. Calculations of the percentage weight retained

Sample code for the calculations of the percentage weight retained is as follows

- for(i in 1:nrow(dd)) {
- a<-as.double(dd\$wt_retained_in_each_sieve_gm[i])

- c[i]<-as.double(a/soilsample)*100.000 }

Similarly the calculations for cumulative percentage weight retained, percentage finer have been calculated.

The calculations part is displayed on the screen as follows

samplename	sieve_size_in_mm	wt_retained_in_each_sieve_gm	percentage_on_each_sieve	cumulative_percentage_retained
1 Raja	37.50	551.00	55.10	
2 Raja	19.00	200.00	20.00	
3 Raja	9.50	100.00	10.00	
4 Raja	4.75	50.00	5.00	
5 Raja	2.36	25.00	2.50	
6 Raja	1.18	12.00	1.20	
7 Raja	0.60	6.00	0.60	

Figure 7: Depicting the display of calculations part

2.4.5. Calculations of the percentage weight retained

Sample code for the calculations of the percentage weight retained is as follows

- ef_1<-dbGetQuery(con,paste0("select sieve_size_in_mm,percentage_finer from example where sieve_size_in_mm='75'"))
- y<-ef_1\$percentage_finer
- ef_2<-dbGetQuery(con,paste0("select sieve_size_in_mm,percentage_finer from example where sieve_size_in_mm='19'"))
- z<-ef_2\$percentage_finer
- if(!is.null(ef_2\$sieve_size_in_mm)&& length(ef_2\$sieve_size_in_mm)>0&&ef_2\$sieve_size_in_mm<75&&ef_2\$sieve_size_in_mm>=19) { rs[i]<- y-z }
- rs[i] <- lapply(rs,round,3)
- percentage_wtretained_on_each_sieve<-do.call("rbind",rs)

Similarly the calculations for percentage weight retained on re-maining sieves have been calculated

The calculations part is displayed on the screen as follows

	groupname	grainsize_mm	percentage_wtretained_on_each_sieve
1	Boulder-size	>300	0.00
2	Cobble-size	80-300	10.00
3	coarse-gravel	20-80	8.01
4	fine-gravel	4.75-20	18.99
5	coarse-sand	2-4.75	7.61
6	medium-sand	0.425-2	22.44
7	fine-sand	0.075-0.425	30.24
8	silt	0.002-0.075	2.70
9	clay	<0.002	0.00

Figure 8: Depicting the display of calculations part

2.4.6. Visualisation of particle size distribution graph

The sample code for the visualisation is as follows

- cl<-dbGetQuery(con,paste0("select sieve_size_in_mm,percentage_finer from example"))
- data_x <- c(cl\$sieve_size_in_mm)
- data_y <- c(cl\$percentage_finer)
- label_x <- 'SIEVE_SIZE(mm)'
- label_y <- 'PERCENT_FINER'
- x<-data_x
- y<-data_y
- DATA <-

```
data.frame(data_x, data_y)
➤ reverselog_trans <- function(base = exp(1)) {
➤ trans <- function(x) -log(x, base)
➤ inv <- function(x) trans_new(paste0("reverselog-",
    for-mat(base)), trans, inv,

➤ log_breaks(base = base),
➤ domain = c(1e-100, Inf) }
➤ trans=reverselog_trans(10)
➤ ppt<-ggplot(DATA, aes(x = data_x, y = data_y)) +
    geom_line()+ scale_x_
        continuous(trans=reverselog_trans(10))+
➤ labs(x = label_x, y = label_y, title = 'SIEVE
    ANALYSIS') + theme_bw(base_size =
        15) +theme(plot.title = element_text(hjust =
        0.5,colour = 'blue'), panel.background =
        element_rect(colour = 'black', fill = "white"),
        panel.grid.major = element_line(colour =
        "#E5E5E5"))
➤ output$plot<-renderPlot(ppt)
```

The particle size distribution graph is displayed as follows



Figure 9: Depicting the particle size distribution for the given soil sample

2.4.7. Predicting of D10, D30, D60 values using uniroot function

The sample code for the calculation of D10, D30, D60 in server.R is as follows

```
➤ x<- c(cl$sieve_size_in_mm)
➤ y <- c(cl$percentage_finer)
➤ spl <- smooth.spline(y ~ log10(x))
➤ newy1 <- 10.00
➤ D10<-uniroot(function(x) predict(spl,log10(x),
    de-ri-ve=0)$y-newy1,interval=c(0,100))$root
➤ newy2 <- 30.00
➤ D30<-uniroot(function(x) predict(spl,log10(x),
    de-ri-ve=0)$y-newy2,interval=c(0,100))$root
➤ newy3 <- 60.00
➤ D10<-uniroot(function(x) predict(spl,log10(x),
    de-ri-ve=0)$y-newy6,interval=c(0,100))$root
```

The calculated values of D10, D30 and D60 are as follows



Figure 10: Depicting the display of D10, D30 and D60 values

2.4.8. Calculation of Cc and Cu values

Sample code of the calculation of Cc and Cu values in server.R

```
➤ cc= (D30*D30)/(D60*D10)
➤ cu= D60/D10
```

The display of the Cc and Cu values is as follows

Cc	Cu
1.88	12.25

Figure 11: Depicting the display of calculated Cc, Cu values

2.4.9. Classification of soil using artificial intelligence

2.4.9.1. Creation of classes and subclasses

The concepts which are mentioned in the mind map are converted into classes and sub classes under thing super class. If there is a tool support for creation of classes and sub classes it can be used. For example in Fluent Editor to write the following code can be used to creation of creation of classes and sub classes.

The sample code for the creation of classes and sub classes is as

- “ Every soil is a sample.
- Every coarse-grained is a soil.
- Every fine-grained is a soil. ”etc

The following figure shows the created classes and subclasses.

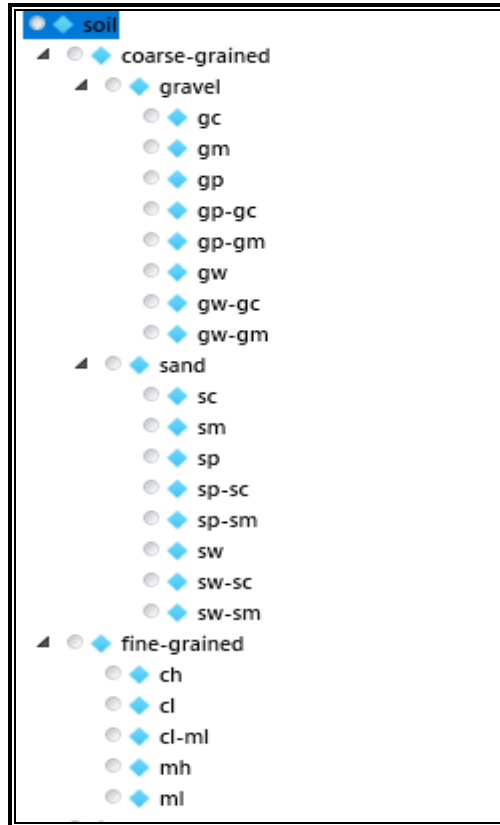


Figure 12: Display the created classes and subclasses

2.4.9.2. Creation of data type properties

Data type properties provide the relationship between one individual and data values. For example sample instance "SAMPLE-1" can have data property `hasPercentageWeightRetainedOn4.75mmSieve` with value as double.

The sample code for the creation of Data Property and assigning the values to the Data Properties is as follows.

- " Sample-1 has-cc equal-to 2.5.
- Sample-1 has-cc equal-to 3.5.
- Sample-1 hasPercentageWeightRetainedOn4.75mmSieve equal-to 55.0." ...etc

The following figure shows the some data properties in soil classification.

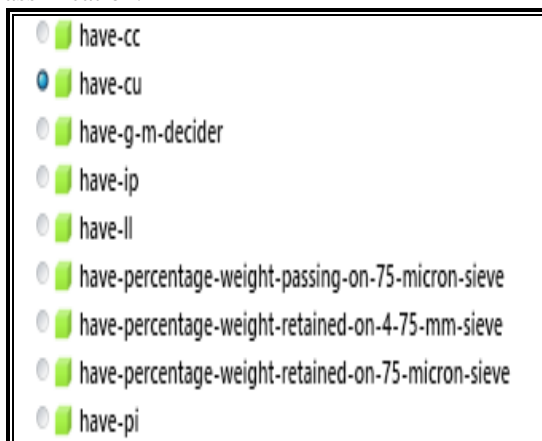


Figure 13: Display the created Data type properties.

The following are the some data properties in soil classification

Sr No	Data Properties in soil classification	
	Data Property	Data type and description
1	<code>hasPercentageWeightRetainedon75MicronSieve</code>	Double & user Input
2	<code>hasPercentageWeightRetainedon4.75mmSieve</code>	Double & user Input
3	<code>Hascc</code>	Double & user Input
4	<code>Hascu</code>	Double & user Input
5	<code>hasLL</code>	Double & user Input
6	<code>hasPL</code>	Double & user Input
7	<code>hasPI</code>	Double & Calculated by rule
8	<code>hasPercentageWeightPassingon75MicronSieve</code>	Double & Calculated by rule
9	<code>hasALineValue</code>	Double & Calculated by rule
10	<code>hasClayDecider</code>	Double & Calculated by rule

Table 2: Some data properties used in soil classification

2.4.9.3. Giving meaning to the concepts

Semantic web is all about giving meaning to concepts. In owl one such provision to give meaning is equivalent. For example the following sample code will explain the meaning in equivalent of class description of Fluent Editor and also the meaning of GW classification of soil can be given.

- " Something is a GW if-and-only-if-it is a gravel and has-cc (greater-or-equal-to 1.0, lower-or-equal-to 3.0) and has-cu greater-or-equal-to 4.0 and hasPercentage-PassingOn75micronSieve lower-than 5.0."

2.4.9.4. Creation of individuals

Under sample class, instances of (SAMPLE-1, SAMPLE-2) are created. By using the following sample code we can create the instances to the sample class.

- " Sample-1 is something.
- Sample-1 is a sample. "

The following figure shows the created sample instances.

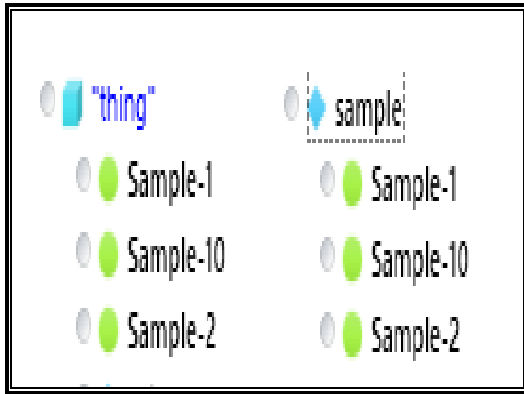


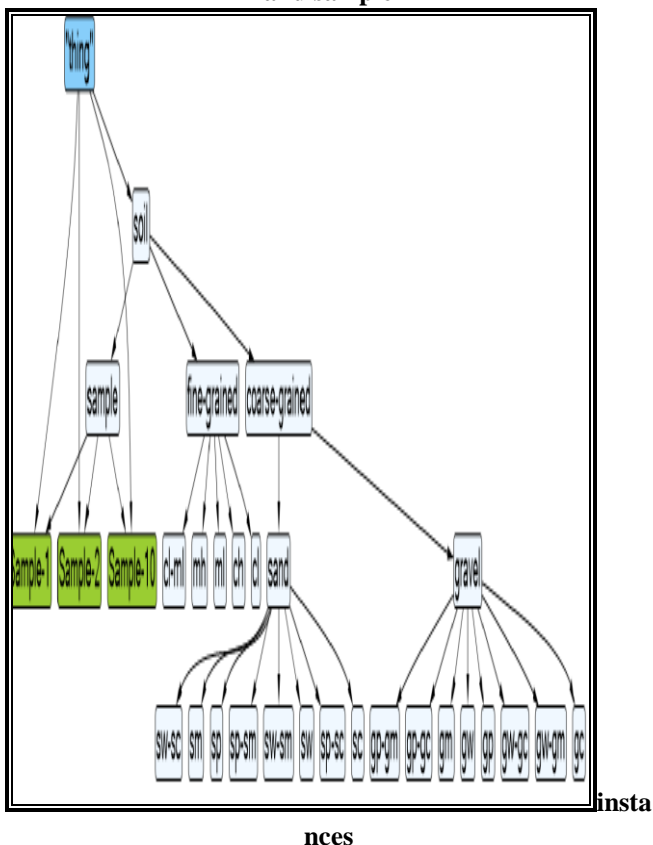
Figure 14: Display the created Individuals

The values for the data type properties can be provided to the sample instances can be as follows in the Fluent Editor.

- “ Sample-1 hasLL equal-to 45.0.
- Sample-1hasPercentageWeightRetainedOn4.75mmSieve equal-to 45.0.
- Sample-1hasPercentageWeightRe-tainedOn75micronSieve equal-to 55.0. ”

The following figure shows the created classes, subclasses and sample instances as CNL Diagram in the Fluent Editor.

Figure 15: Display of the created classes, subclasses and sample



The following figure shows the created classes, subclasses and sample instances as in the Fluent Editor.

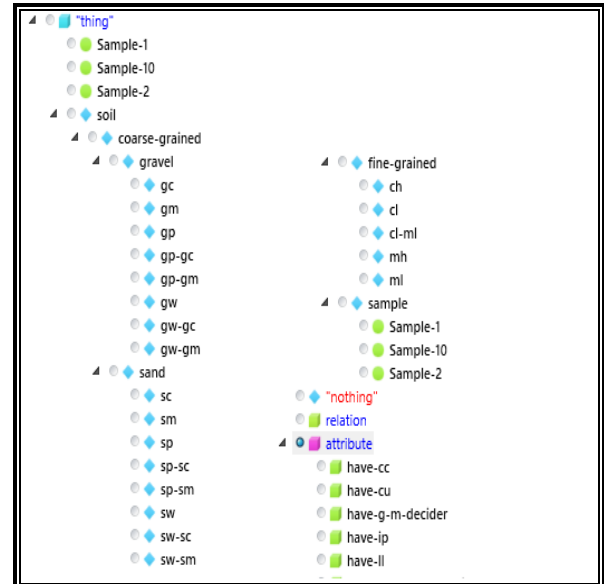


Figure 16: Display the created classes, subclasses and sample instances

2.4.9.5.Axioms in rOntorion

After creating the ontology syntaxes in the Fluent Editor select all ontology syntaxes and then paste it in the R code by using the package rOntorion as follows.

The sample code for the ontology syntaxes in the R code is as follows.

- axm1<-paste0("Sample-1 is something.")
- axm2<-paste0("Sample-10 is something.")
- axm3<-paste0("Every fine-grained is a soil.")
- axm4<-paste0("Something is a fine-grained if-and-only-if-it is a soil and has-percentage-weight-retained-on-75-micron-sieve lower-or-equal-to 50.0.") , etc.

2.4.9.6. Database

The values for the data type properties can be provided to the sample instances from the database will be as follows in the R code.

The sample code for the ontology syntaxes in the R code to provide values to instances from the database is as follows.

- axm110<-paste0("Sample-2 has-cc equal-to ",cc,".")
- axm111<-paste0("Sample-2 has-cu equal-to ",cu,".")
- axm112<-paste0("Sample-2 has-ll equal-to ",ll,".") , etc

2.4.9.7. Load otology in R code

The all axioms will be stored to string. And then load the string to the R code by using rOntorion package code.

The sample code for load the axioms to the R are as follows.

- str1<-paste(axm1, axm2, axm3, axm4, ... , axm110, axm111, axm112, sep="\n")

- onto<-ontorion.load.cnf.string(str1,TRUE) or
- onto<-ontorion.load.rdf.string(str1,TRUE)

2.4.9.8. Querying ontology

The loaded ontology string and query to the required instance will be used to call the ontology query.

The sample code for the Querying the ontology is as follows.

- some<-paste0("something that is Sample-1")
- result1<- ontorion.direct.instances.of(onto, some)
- or
- result1<- ontorion.direct.super.concepts.of(onto, "Sam-ple-1")

The following figure displays the result of the soil classification.

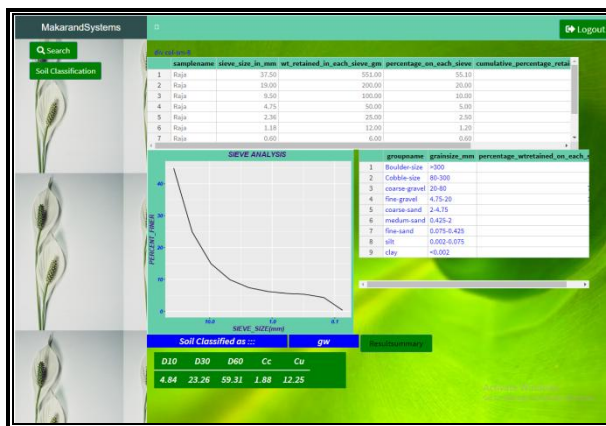


Figure 17: Display of the classification of soil results

2.4.9.9. Conversion of ontology to .rdf format

The sample code for the conversion of ontology code to .rdf format is as follows

- str12<-paste(axm1, axm2, axm3, axm4, axm5, axm6, axm7, axm8, axm9, axm10, axm11, axm12, axm13, sep="\n")
- onto1<-ontorion.load.cnf.string(str12,TRUE)
- print(ontorion.as.rdf(onto1))

The following figure shows the conversion of code in .rdf format

```
<?xml version = '1.0' encoding = 'UTF-8'?>\r\n<rdf:RDF
xmlns=\ "http://www.cognitum.eu/onto#"
xml:base=\ "http://www.cognitum.eu/onto"
xmlns:owl=\ "http://www.w3.org/2002/07/owl#"
xmlns:rdfs=\ "http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf=\ "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:xsd=\ "http://www.w3.org/2001/XMLSchema#">\r\n
<owl:Ontology rdf:about=\ "http://www.cognitum.eu/onto#"
/>\r\n <!-- \r\n http://www.cognitum.eu/onto#hasCc -->\r\n
<owl:DatatypeProperty
rdf:about=\ "http://www.cognitum.eu/onto#hasCc" />\r\n
<!-- http://www.cognitum.eu/onto#hasCu -->\r\n
<owl:DatatypeProperty
rdf:about=\ "http://www.cognitum.eu/onto#hasCu" />\r\n
<!-- http://www.cognitum.eu/onto#hasGMDecider -->\r\n
<owl:DatatypeProperty
rdf:about=\ "http://www.cognitum.eu/onto#hasGMDecider"
/>\r\n <!-- http://www.cognitum.eu/onto#hasLl -->\r\n
```

```
<owl:DatatypeProperty
rdf:about=\ "http://www.cognitum.eu/onto#hasLl" />\r\n
<!--
http://www.cognitum.eu/onto#hasPercentageWeightPassing
On75MicronSieve -->\r\n
```

III. ABBREVIATIONS

AI: Artificial Intelligence
Owl: Web Ontology Language
SWRL: Semantic web rules language
RDF: Resource Description Framework
GW: Well graded Gravel
Cu: Coefficient of uniformity
Cc: Coefficient of curvature
TBM: Tunnel Boring Machine

IV. FUTURE WORK

At present rOntorion cannot be deployed in Linux web server, where as shiny web server needs Linux operating system.

V. CONCLUSION

Soil classification is the basic requirement for any geotechnical engineer need to know before embarking on any construction project. With the advancement in the information technology it is possible for the machines to understand the meaning of the engineering concepts using semantic enabled applications.

An innovative approach for development of artificial intelligence enabled web application by taking use of the visualisation features of R, development of the user friendly application using shiny web server, rhandsonable and rOntorion for development of owl on-tology is presented.

It is found that semantic enabled artificial intelligence application for engineering classification of soils can be developed using the R and OWL technologies where particle size distribution curve can be visualised, D60, D30, D10 values can be predicted and soil classification can be performed so that machines and humans can interact..

Acknowledgement

Thank the senior director of Makarand systems pvt limited, P.Ramesam for funding the project and providing the facilities in order to complete the project

REFERENCES

1. Indian Roads Congress, "Quality Assurance Handbook For Rural Roads", Vol -1, pp31, May 2007
2. Stephen Balkirsky et al., "Towards A Robot Task Ontology Standard", Proceedings of ASME 2017 International Manufacturing Science and Eng. Conference, MSEC 2017 June 4-8, 2017, Los Angeles, CA, USA
3. Pankesh Patel et al., From raw data to smart manufacturing, IEEE Intelligent Systems, Volume 33, Issue 4
4. Jimi Condilite, "Next Up in Driverless Vehicles: Autonomous Excavators". Technology Review, <https://www.technologyreview.com/the-download/609174/next-up-in-driverless-vehicles-autonomous-excavators/>, Oct 19, 2017
5. G. Kandaswamy, Ministry of road transport and highways specification for road and bridge works, Indian Roads Congress, Fifth revision, pp62, Jan 2013
6. Deccan Chronicle, "At last: Godavari TBM's tryst with Bengaluru's majestic station" <https://www.deccanchronicle.com/nation/current-affairs/200416/at-last-godavari-tbm-s-tryst-with-bengaluru-s-majestic-station.html>, 20 Apr 2016
7. Kaklamanos, KT Elmy, Development of a Geotechnical Engineering Software Package in R and Its Implementation in the Civil Engineering Curriculum, Geotechnical and Structural Engineering Congress 2016, 635-647.
8. R.G. Raskin and M.J. Pan. Knowledge representation in semantic web for earth and environmental terminology (SWEET), computers & geosciences, 31(9):1119-1125, 2005.
9. M. Zhao, Q. Zhao, D. Tian, P. Qian, and X. Zhang. Ontology-based intelligent retrieval system for soil knowledge. WSEAS Transactions on Information Science and Applications, 6(7):1196-1205, 2009.
10. T. Heepaisong and A. Shivihok. Soil Knowledge-based Systems Using Ontology. In Proceedings of the International Multi-Conference of Engineers and Computer Scientists, pages 1-5, 2012.
11. P. L. Buttigieg, N. Morrison, B. Smith, C. J. Mungall, and S. E. Lewis. The environment ontology: contextualising biological and biomedical entities. Journal of Biomedical Semantics, 4:43, 2013.
12. P. Shivananda and P. Srinivas Kumar. Building Rules Based Soil Classification Ontology. International Journal of Computer Science and Information Technology & Security, 3(2), 2013.
13. C. Deb, S. Marwaha, P. Malhotra, S. Wahi, and R. Pandey. Strengthening soil taxonomy ontology software for description and classification of USDA soil taxonomy up to soil series. In Proceedings of the 2nd International Conference on Computing for Sustainable Global Development, pages 1180-1184, 2015.
14. H Du, V Dimitrova, D Magee, R Stirling, G Curioni, H Reeves, B Clarke. An ontology of soil properties and processes, International Semantic Web Conference, 30-37
15. D.H. Deng, Z.Y. Gong, Z.L. Guo and S. Phillip. (2008). Semantic programming of Web-enabled database applications, Proceedings of 1st IEEE International Workshop on Semantic Computing and Applications, IEEE Press, 2008, 51-60.
16. A. Sheth, "Internet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing," IEEE Intelligent Systems, vol. 31, no. 2, 2016, pp. 108-112.
17. I. Grangel-González et al., "The Industry 4.0 Standards Landscape from a Semantic Integration Perspective," Proc. 22nd IEEE Int'l Conf. Emerging Technologies and Factory Automation (ETFA), 2017, pp. 1-8.
18. A. Gyrard et al., "Building the Web of Knowledge with Smart IoT Applications," IEEE Intelligent Systems, vol. 31, no. 5, 2016, pp. 83-88.
19. Shiny server Introduction, Jeff Allan, <https://www.rstudio.com/products/shiny/shiny-server/>, 25-Feb-2014
20. PostgreSQL Introduction <https://www.postgresql.org/about/>
21. Ontorion Introduction <http://www.cognitum.eu/semantics/FluentEditor/rOntorionFE.aspx>