M3 framework specifications

Functionnalities :

## M3 converter

Input: sensor data

Output : semantic sensor data

* + Get senML/XML
  + Get SenML/Json
  + From the grey interface
  + From the raspberry
  + Query web service
  + Read file
  + According to the m3 ontology
  + Gerer les units – compliant with m3
  + Gerer les measurements – compliant with m3
  + Gerer les domains – compliant with m3
  + Ecrire la valeur m3:hasValue
  + Avec les Jena rules
  + Ou directement dans le code java ?
  + convertSenMLToRDF
  + convertToM3WithJenaRules (exectute M3 rules + reasoner)
  + convertToM3WithJava
* Utils
  + Load file in a string
  + Load file in a jena model
  + Query web service
  + Load triplestore
  + Update triplestore
  + Update load database tdb
* Reasoning
  + Infer new data and return data inferred
  + String, file or model jena
  + Ecrire dans le fichier les nouvelles data RDD/XML
  + Les mettre a disposition dans un triple store
  + Les metttre a disposition via le sparql endpoint fuseki
  + Load rules
  + Execute reasoning engine
* Query data
  + Crete une requete sparql
  + Execute requete sparql
  + Replace variables
  + Get result
  + Parse result
  + return result
  + integrate linked data api pour generic sparql
* build new application
  + give sensor
  + give domain
  + give sensor data
  + find ontologies
  + find dataset
  + find sparql queries + variables
  + apply reasoner on sensor data
  + return result parsed
* link domain
  + integrate onto mapping
  + extract onto extract rules
  + linked rules
  + linked onto
  + rewrite into m3 rules
  + rewrite into m3 onto

junit

scenario

S-LOR  
updata body temperature

Test wil all values

Precipitation

Etc.