* 1. Overview

\* For developing a software a requirement list is given.

\* A software developer designs the software using various classes, functions etc.

\* From this software we can obtain an ontology.

\* The ontology is then matched with the requirement list and a mapping between these two is created

\* We determine quality measurements such as consistency, completeness, correctness, unambiguity.

\* The aim of this research is to improve these measurement values using ontology

* 1. Requirements

Codeontology – a parser to make rdf triples from java src code and jar files

Protégé – a software to build and visualize ontology

* 1. current approach:

One of the most crucial problems to automate requirements analysis is that requirements documents are usually written in natural language

e.g. English or Japanese. Although techniques for natural

language processing (NLP) are being advanced nowadays,

it is hard to handle such requirements documents sufficiently

by computer

again its usage is very limited to practitioners because of their difficulty and complexity in learning and training(for devs).

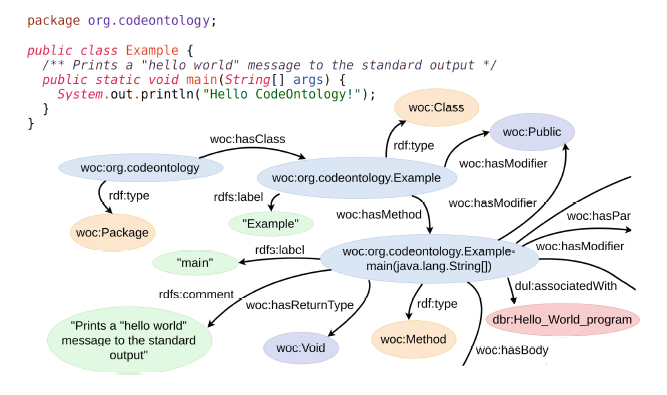
This is why automatic semantic analysis with light weight pre-processing is more practical in requirement analysis

* 1. Our approach to solve the problem

1. Build an ontology from s/w

Ontology: a formal representation of oop lang(place it in a diff slide, with a fig.). classification of the existing concepts (lightweight…)

Rdf triples



1. Mapping the ontology with the given requirement list

* Requirement list : A list of itemized sentences written in natural language. each sentence represents a customers

requirement to a software system to be developed

* Mapping : a relationship between a concept from the requirement list and a concept from the ontology

1. Measuring completeness, correctness, unambiguity in this mapping
   1. Related theories

P(x): the type of a concept or a relationship “x” is “P”.

• R(x,y): the type of relationship between concept “x”

and “y” is “R”. e.g., aggregate(car, tire), generalize(

operation, play), apply(delete, file), require(player,

codec).

• InSpec(x,S): requirements document S refers to concept

or relationship x. e.g., InSpec(A, S) and In-

Spec(require(A,C), S) are true but InSpec(C, S) is false

in Figure 1.

ontology

system can be represented in the following way.

Ontology System = (Con, Rel, Rules)

Con: a set of concepts. Con = ∪t∈TypeCont

Cont: a set of concepts where its type is t, e.g.,

Cont=“function”

Rel: a set of relationships

Concept: quality, function, object, environment, constraint

Quality analysis:

Correctness (COR): The requirements items that were

mapped into ontological elements can be considered

as requirements appropriate for a problem domain.

COR =

# *{*requirements items that are mapped into

the ontology*}*

# *{*requirements items (total number of requirements

items)*}*

2. Completeness (CMP) : The ontological elements that

did not have any mapped requirements items can be

candidates for missing requirements items.

CMP =

# *{*ontological elements that no requirements

items are mapped into*}*

# *{*ontological elements (total number of ontological

elements)*}*

3. Consistency (CST) : If requirements items include

ontological concepts that are connected through

“contradict” relationship in the ontology, they are

inconsistent.

CST =

# *{*ontological relationships that are not

“contradict” and that some requirements

items are mapped into*}*

# *{*ontological relationships that some requirements

items are mapped into*}*