Deliverable number:	D1.3
Deliverable Title:	Action ontology and object categories
Type (Internal, Restricted, Public):	PU
Authors:	Daiva Vitkute-Adzgauskiene, Jurgita Kapociute, Irena Markievicz, Tomas Krilavicius, Minija Tamosiunaite, Florentin Wörgötter
Contributing Partners:	UGOE, VMU

Project acronym:	ACAT
Project Type:	STREP
Project Title:	Learning and Execution of Action Categories
Contract Number:	600578
Starting Date:	01-03-2013
Ending Date:	30-04-2016

Contractual Date of Delivery to the EC:	31-08-2015
Actual Date of Delivery to the EC:	04-09-2015

Content

1. Executive summary

This deliverable provides a report on the structure of the action ontology developed for the ACAT project. The structure of the ontology is motivated by the needs of the instruction textual compiler (reported as such in D3.1). The use of the created ontology in the work of the compiler is explained.

The ACAT ontology is organized based on WordNet lexical ontology principles, indicating hierarchies of actions and objects. The most important aspect of the ACAT ontology not present in the WordNet ontology is the connection between action names (verbs) and objects (nouns) thus forming a so called "action environment" for each action.

The ACAT ontology is kept domain specific, only including those WordNet synsets which are present in ACAT domains (chemical laboratory and industrial assembly). However, the ACAT ontology and the WordNet are compatible and when needed the ACAT ontology can be used together with the WordNet ontology. In addition, the ACAT ontology has elements that are not existing in the WordNet. These are mostly multiword object names from ACAT scenarios (like "rotor-shaft" or "measuring beaker").

2. Introduction

The ACAT ontology is built with the aim of organizing and structuring language level action information. Each action is characterized by a corresponding action word (verb) and, also, by action-bound object names, specifying which objects are relevant to the action and which roles they perform. Action related objects form the context of an action in the sentence. In the ontology action-related objects we call action environment.

Both action and object names in the ontology are organized in a hierarchical structure, following hypernym – hyponym hierarchies typical to the WordNet lexical ontology of the English language (https://wordnet.princeton.edu/).

The ACAT ontology is used by the ACAT textual instruction compiler for identifying object roles in the instruction as well as for filling in information that is missing in the instructions. Instructions written for humans often lack information which can be attributed to commonsense knowledge, including information about tools or other action-context related objects. In case of several possible results when querying the ontology for missing action-context information, ranking is used for picking out the most probable set of information. This ranking is mainly based on the object roles in the action context.

Object related information in the ACAT ontology is structured by categorizing objects by their semantic roles in the instruction sentence.

3. Overview of the ACAT ontology structure

The general structure of the ACAT ontology is presented in Fig. 1. Two main ACAT ontology classes (ACTION and OBJECT) determine the hierarchical structure of action hypernyms/troponyms and object hypernyms/hyponyms (property subClassOf). Each action and object synset contains a subset of synonymous instances (property instanceOf) having the same definition as their parent class. The subclasses of ACTION class are described by the following properties: main_action, robotic_action, supportive_action (See D1.2 Chapter 4 for explanation of the properties). The subclasses of OBJECT class are defined by the property part_of to describe object

holonym/meronym relations. Also, each action and object can be described by the values of annotation properties from WordNet and corresponding instruction sheets (label, gloss and example). The relation between an action and an object is determined by the following restriction properties: with_tool, with_main_object, with_primary_object and with_secondary_object (the meaning of the "main object", "primary object", "secondary object" are explained in ACAT Term Glossary provided in PPR2, as well as in D1.2 Chapter 2).
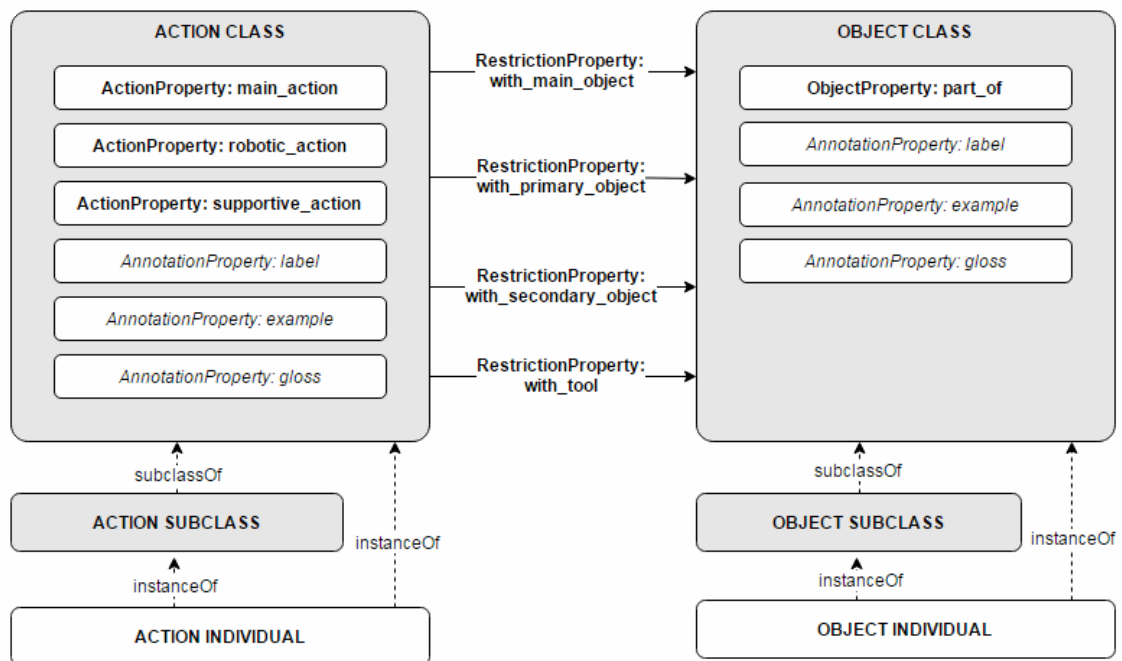

Fig. 1. The structure of the ACAT ontology

The ACAT ontology is formed by assigning an appropriate action synset for each action and, also, all action details required for action execution which form action environment in the ontology. Here, a synset is a synonym ring, which groups semantically equivalent data elements. An action *synset* contains verbs, prepositional verbs (verb + preposition), phrasal verbs (verb + adverb) and other multiword verbs, having the same sense. A verb usually has more than one sense and its' sense can change in collocation with other words, e.g. a direct object name, a preposition or a

certain modifier (e.g. *don't*). E.g., the following verbs can be marked as synonyms: *put in = put into, put out = put away*.

Following this synset approach, the following examples from the ACAT ontology can be given: the synset for the "bring" action consists of the following members: "bring", "convey", "fetch", "get"; the synset for the "raise" action consists of the members: "bring up", "elevate", "get up", "lift", "raise" (Fig. 2). Such an approach allows the textual instruction compiler to handle situations of synonym action names used in the instruction text. All action synsets are members of ACTION ontology class in the ACAT ontology. Actions are classified to main, robotic and supportive actions (D1.2 Chapter 4) by adding corresponding properties in the ACAT ontology.
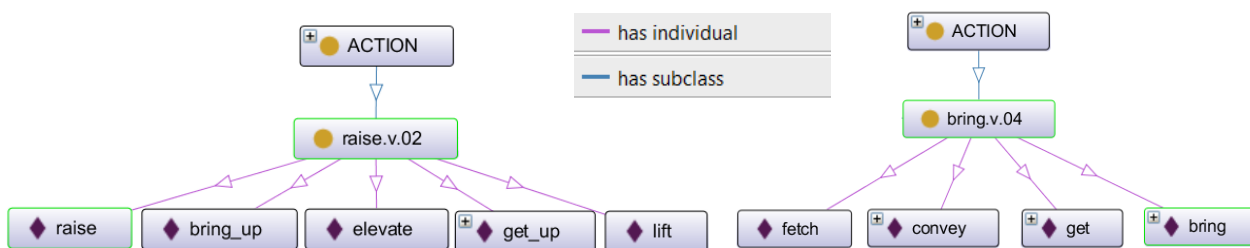

Fig. 2. Examples of action classes and their entities

Action environment description includes necessary elements for robot activity: involved tools and materials, main, primary and secondary objects (as they were introduced in the ADTs for robotic action description, see D1.2 Chapter 2), etc. In the ACAT ontology structure they are represented by corresponding object synsets. An object *synset* contains nouns or multiword expressions (usually consisting of adjectives and nouns) having the same sense. For example, the synset for the "conveyor" object consists of the following members: "conveyor", "conveyor belt", "transporter", while the "belt" is a superclass (hypernym) of the "conveyor" (Fig. 3). Such an approach allows the textual instruction compiler to handle situations of synonym object names used in the instruction text. All object synsets are members of the ontology class OBJECT in the ACAT ontology.
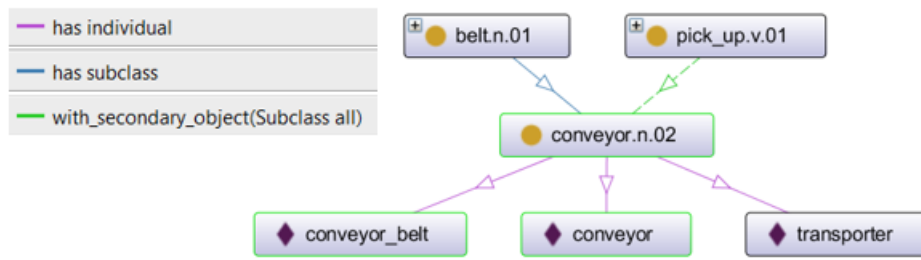
Fig. 3. Example of *conveyor* class hierarchical structure and class entities

The objects in action environment synsets are grouped by their semantic roles, e.g. main, primary, secondary objects, etc. (see Section 4). This is implemented by inserting corresponding object properties when linking actions and objects in the ACAT ontology. These properties include: "with main object", "with primary object", "with secondary object" and "with tool". For example, the "move" action synset, consisting of the members "go", "locomote", "move" and "travel", is linked with the primary objects (object synsets) "platform", "station", "table", and with the secondary objects (object synsets) "axle", "box", "dispenser", "platform", "table", etc. (Fig. 4). Such a linking approach allows to handle different types of links between actions and objects – for example, a "table" object can have the "primary object" or the "secondary object" role, depending on the instruction sentence.
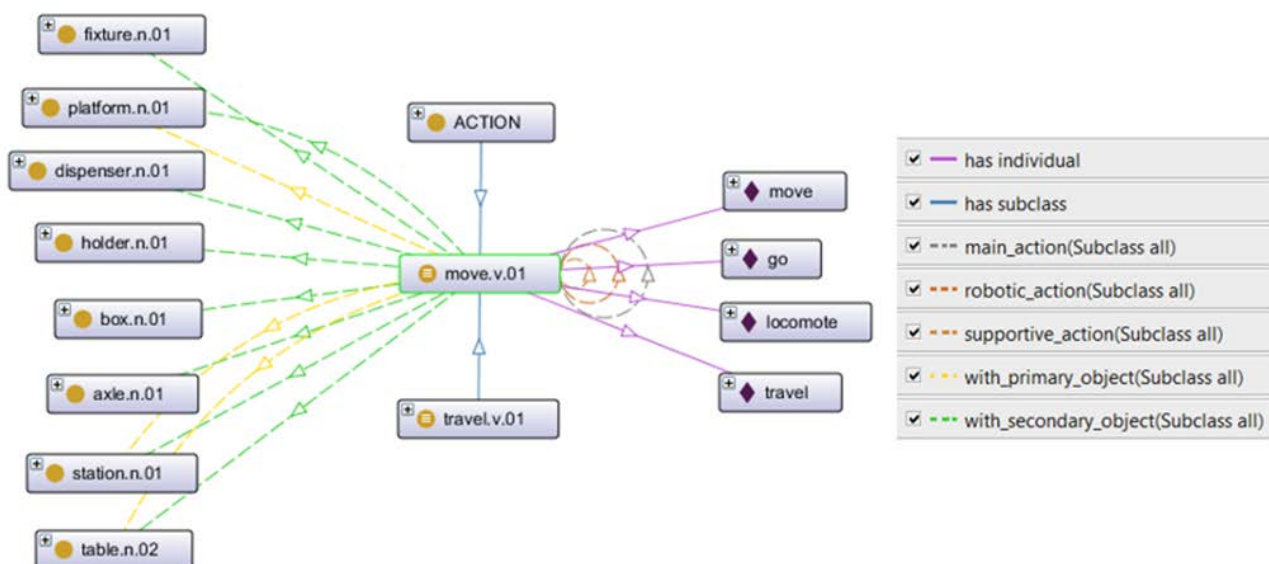

Fig. 4. Example of "move" action relations to objects

Both action and object synsets in the ACAT ontology are organized into hierarchical structures following the hierarchical concept model in the WordNet ontology (see Section 5). The hierarchical structure is also defined by adding property *part_of*, which describes object holonym/meronym relation. For example, the synset "rotor axle" is a subclass of the synset "axle" – also, it is a part of "rotor shaft" (Fig. 5).
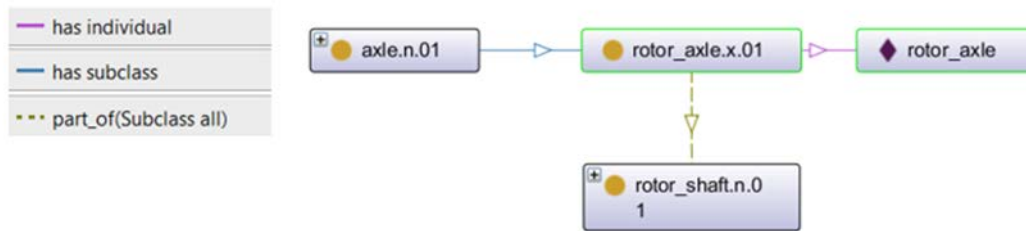

Fig. 5. Example of *part_of* relation between "rotor axle" and "rotor shaft"
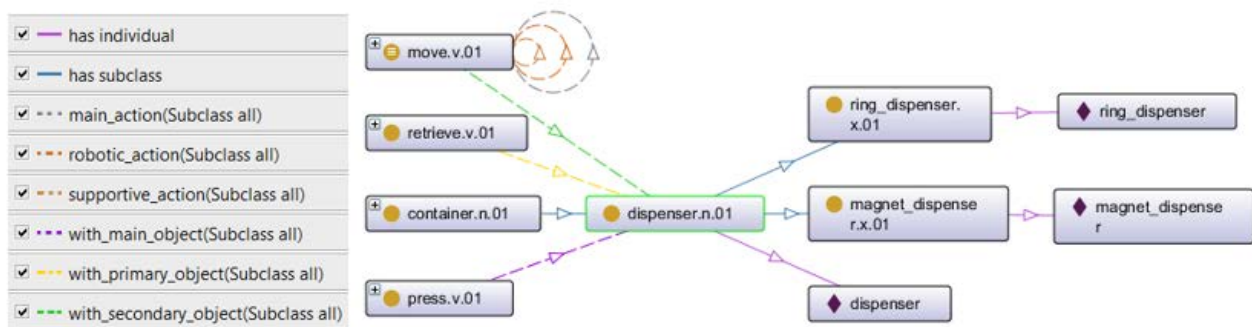

Fig. 6. Example of object "dispenser" definition by its relation to executed actions

The example in Fig. 6 presents the detailed information about object "dispenser": its hypernyms ("container") and hyponyms ("magnet dispenser", "ring dispenser"). Also, it determines relations to executed actions: the action "move" from the instruction sheets was executed with the secondary object "dispenser", the action "retrieve" with the primary object "dispenser", action the "press" with the main object "dispenser". The detailed usage of the object "dispenser" is described in Fig. 7

Synset name	Detailed usage	Description

Dispenser_synset_3177504	Found 8 uses of dispenser_synset_31775
	▼ ◆ **dispenser**
	   ◆ dispenser **Type** dispenser_synset_3177504
	
	▼ ● **dispenser_synset_3177504**
	   ● Class: dispenser_synset_3177504
	   ● dispenser_synset_3177504 **SubClassOf** container_synset_3063333
	
	▼ ● **magnet_dispenser_synset-obj7**
	   ● magnet_dispenser_synset-obj7 **SubClassOf** dispenser_synset_3177504
	
	▼ ● **move_synset_1818343**
	   ● move_synset_1818343 **SubClassOf** with_secondary_object **exactly** 1 dispenser_synset_3177504
	
	▼ ● **press_synset_1434568**
	   ● press_synset_1434568 **SubClassOf** with_main_object **exactly** 1 dispenser_synset_3177504
	
	▼ ● **retrieve_synset_2226903**
	   ● retrieve_synset_2226903 **SubClassOf** with_primary_object **exactly** 1 dispenser_synset_3177504
	
	▼ ● **ring_dispenser_synset_obj6**
	   ● ring_dispenser_synset_obj6 **SubClassOf** dispenser_synset_3177504
		Object synset,
		
		Gloss: a container so designed that the contents can be used in prescribed amounts

Fig. 7. Detailed information for the "dispenser" object from the focused ACAT ontology

The ACAT ontology allows more detailed action environment descriptions by establishing links between the ontology data collection consisting of an action and a set of environment objects on one side and a so-called ADT table on the other side. Fig. 8 presents the conceptual algorithm, describing the process of appending missing object information using the ontology data, where object name is extracted from parsed instruction. The sentence "Take a rotor cap and place it on fixture" contains two action verbs: "take" and "place". By querying the ontology, the system gains knowledge, which of these two verbs takes the "main action" role. Also, the parsed instruction does not include information about the primary object for the action (the position of main object) – ontology data allows to resolve this issue and gives two possible positions of the main object: "table" and "conveyor". If there is no possibility to find the object, which was defined in the instruction sheet, additional data about object hypernyms/hyponyms and holonyms/meronyms can be used.
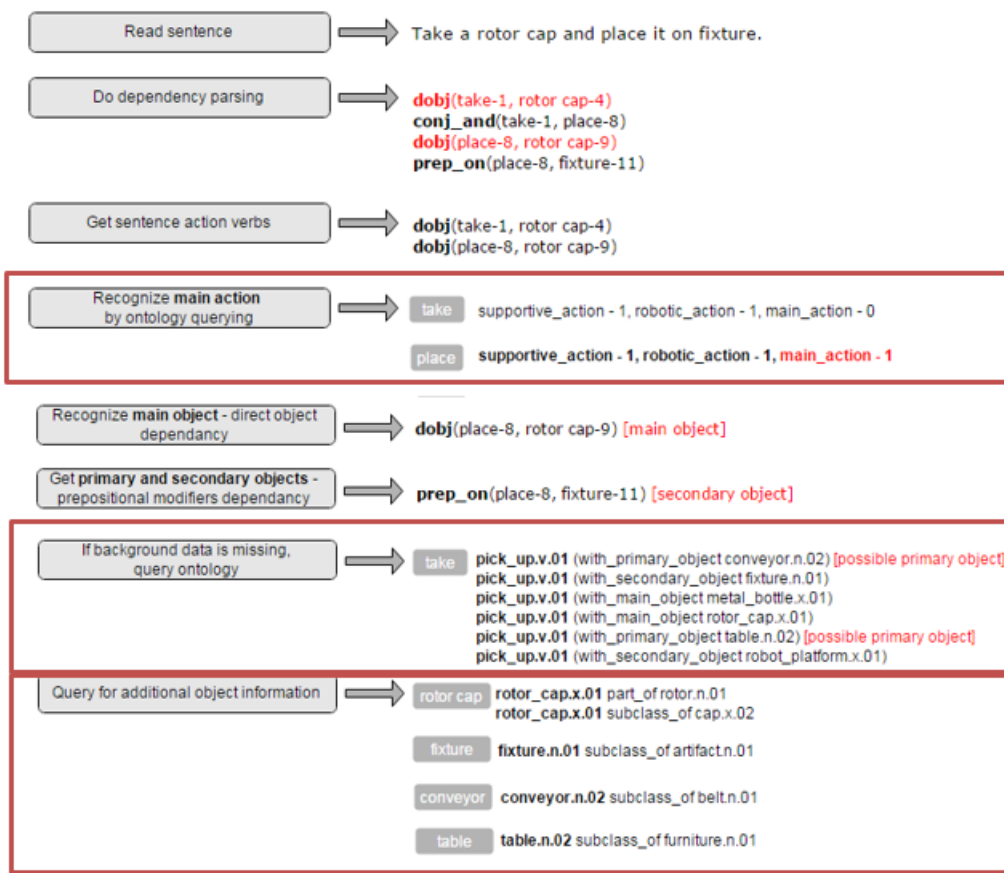
Fig. 8. Example of the use of ontology data in ADT filling

4. Object categorization by their semantic roles in the instruction sentence

Four different object types are represented in the ACAT ontology. These object types are predefined by their semantic roles in instruction sentences:

Main Object (MO): The object which is first touched by *hand/tool* (always present).

Primary object=Source (PO): The object which is first touched/untouched by the *main object* (not always present).

Secondary object=Target (SO): The object which is second touched/untouched by the main object (not always present).

Tool: The entity grasped by the hand to perform an action instead of the hand (not always present).

The ACAT ontology is gradually filled by adding new robot specific actions and action environment objects, extracted from domain-specific corpus texts, mainly domain-specific instruction sets and manuals. When adding object information to the ontology, the context of the object name in corpus texts (a bag of neighboring words) is the initial data for defining object semantic roles.

Categorization of the action environment objects according to their action-specific roles is accomplished by selectively applying two scenarios:

1) Action environment object categorization using rules and search patterns.
2) Using a classification algorithm (Support Vector Machine (SVM) method).

When categorizing the action environment objects using rules and search patterns, one source of information is the VerbNet lexicon with a structured description of the syntactic behavior of verbs. Alternatively, syntactic parse trees for instruction sentences are used. When applying automated extraction of rules from VerbNet lexicon database, mapping of the VerbNet thematic roles to the elements of the predefined conceptual action context model used in ACAT is done. The rules are extracted from the VerbNet syntactic and semantic frames for corresponding verbs [2].

In the example with the "*place*" verb (see Table 1), we obtain two possible search patterns, which are indicated in the first column of the table. E.g. in the first case the search pattern *NP V NP PP.DESTINATION* means that we are analyzing the pattern (sequence of parts of speech) noun-verb-noun-preposition-destination. The second column shows which roles are assigned to the analyzed part of speech sequence: first noun means the agent of the action, the second noun means the theme of the action, while the preposition means the destination (here the rolesagent, theme, destination are as given in VerbNet). The third column provides predicates associated to the action where e.g. predicate "**motion**(during(E), **Theme**)" means that we are talking of motion of the object with the role "Theme" over time E. One can observe that the predicates for the two indicated search patterns for verb "place" provided in the table are the same. Defined patterns by their semantics means, that an *Agent* places a *Theme* at a *Destination* - the *Theme* is under the control of the *Agent/Cause* at the time of its arrival at the *Destination*.

These patterns are applied to a morphologically annotated domain specific corpus for filling the action ontology with classified action environment elements. The role "Theme" form the VerbNet

is assigned to be the *main object* in ACAT role notation and the VerbNet role "Destination" is defined to be the *secondary object* in ACAT notation.

Table 1. VerbNet syntactic and semantic frames for verb „*place*" (Source: VerbNet)

Description	Syntax	Semantics	Example
NP V NP PP.DESTINATION	**Agent-NP** (putter) **V Theme-NP** (thing put) {{+loc}} **Destination-PP** (where put)	**motion**(during(E), **Theme**) not(Prep(start(E), **Theme**, **Destination**)) Prep(end(E), **Theme**, **Destination**) cause(**Agent**, E)	***Place*** *the* ***rotor cap*** *on the* ***fixture***.
NP V NP ADVP	**Agent-NP** (putter) **V Theme-NP** (thing put) **Destination** (where put)<+adv_loc>	**motion**(during(E), **Theme**) not(Prep(start(E), **Theme**, **Destination**)) Prep(end(E), **Theme**, **Destination**) cause(**Agent**, E)	***Place*** *the* ***rotor cap*** *here.*

In the ACAT ontology we have included the "tool" category for objects (Fig. 9). Objects belonging to this category can potentially be used as tools in robotic actions. We assign objects to this category using classification algorithms for object name occurrence in corpora texts. Specifically, for classification we use the Support Vector Machine (SVM) method based on the bag of words approach [3]. This method can effectively cope with high dimensional feature spaces, sparseness of the feature vectors and instances not sharing any common features (very common for short texts).
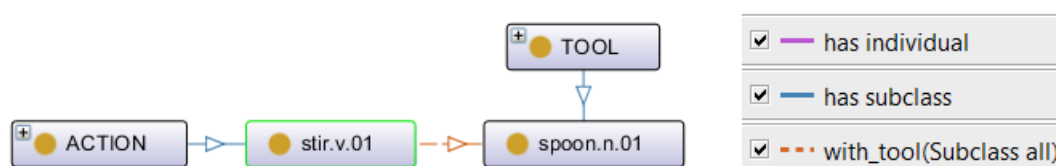

Fig. 9. Example of the use of TOOL in ontology. Relation between action and object is described via "with_tool" restriction.

Though we have not integrated identification of the tool in the instruction compiler, due to the reason that instructions requiring tools were missing in the ACAT instruction sheets provided in D5.1, tool usage potentially may be needed for table top operations in the context of chemical laboratory or industrial assembly (e.g. stir with the spoon). Thus, we have developed the

algorithmic framework required for tool identification. Specifically, we have investigated what feature vectors best describe the object context for object categorization in text. We were considering using n-grams and incorporating part-of-speech (POS) information into features. Table 2 presents a summary of the feature types, which we have investigated in the extraction of object classes by classification of domain specific texts.

Table 2. Feature type groups and feature types (with their description) used in our experiments

Feature group	Feature type	Description
Symbolic	Document-level character n-gram (*chrn*)	Succession of *n* characters including spaces and punctuation marks. We investigated sliding window of $n \in [3; 7]$. E.g. if *n*=5 phrase "verb context" would be split into "verb_", "erb_c", "rb_co", "b_con", "_cont", "conte", "ontex", "ntext".
Lexical	Bag-of-words (*bow*)	N-grams (interpolation of *n* from 1 up to 3) based on word tokens. E.g. if *n*=1 "object context classification" would be transformed into single words "object", "context", "classification"; if *n*=2 it would be transformed into singe words plus pairs of words "object context", "context classification"; if *n*=3 it would be transformed into single, pairs of words plus triplets of words "object context classification".
	Lemmas (*lem*)	N-grams (interpolation of *n* from 1 up to 3) based on word lemmas. All texts were lemmatized beforehand. Lemmatization transformes words into their main form, not changing the part-of-speech tag, e.g. "better" $\rightarrow$ "good", etc.
	Stems (*stem*)	N-grams (interpolation of *n* from 1 up to 3) based on word stems. All texts were stemmed beforehand. Stemming reduced inflected words to their stem: e.g. "friendly" $\rightarrow$ "friend", etc.
Morphological	Part-of-speech tags (*pos*)	N-grams (interpolation of *n* from 1 up to 3) based on part-of-speech tags. All texts were part-of-speech tagged beforehand. For part-of-speech tagging, as well as for lemmatization and stemming Stanford parser ([3]) was used.
Aggregated: Lexical + Morphological	Lemmas + part-of-speech tags (*lempos*) Stems + part-of-speech tags (*stempos*) Bag-of-words + part-of-speech tags (*bowpos*)	N-grams (interpolation of *n* from 1 up to 3) based on aggregated features which involved concatenated lexical and syntactic information. E.g. "filtration_NN" is an example of a *lempos* feature, where "filtration" is a word lemma and "NN" is a part-of-speech tag for determining singular nouns.

Experimental results confirm that the context on the right (i.e., following words) was the most informative and gave the biggest boost in accuracy compared to the context on the left or lying in both directions [1]. The assumption that the best results should be obtained with a relatively small window was confirmed as well: the best results were obtained with 25 symbols (~5 words) only using bag-of-words as features.

5. Hierarchical object structures

Action and object information in the ACAT action ontology is structured by adding paradigmatic information from WordNet lexical database:

- *Hypernym words*. Hypernym is a linguistic term for a word whose meaning includes the meanings of other words. In this case, the hierarchical hypernym structure was limited by 1 level in order not to overload the ACAT action ontology with excess information. In case this information becomes necessary for some specific tasks, it can be extracted from WordNet by using special webservices designed in this project.
- *Hyponym words* where applicable. A hyponym word is a word whose semantic field is included within that of another word.
- *Troponym words* where applicable. Troponym is a verb that indicates more precisely the manner of doing something by replacing a verb of a more generalized meaning.
- *Holonym/meronym* words where applicable. Holonym defines the relationship between a term denoting the whole and a term denoting a part of, or a member of, the whole. Holonym is the opposite of meronym.

The ACAT action ontology structure is formed following the basic structural principles of the WordNet lexical ontology. That is, the action and object entities in the ontology are displayed as individuals of action or object synsets (class members) and each synset class is described by gloss and examples. The examples in case of the ACAT ontology, are taken from corresponding instruction sentences in domain-specific corpus.

Also, the same naming principles for synsets are used both in the ACAT ontology and in WordNet. For example, the ACAT ontology synset "axis.n.06" corresponds to a synset with the same name in

the WordNet ontology. The synset name structure shows that the word is "axis", its POS (part-of-speech) classification is "noun" (n), and this particular meaning of the word "axis" is No.6 in the sequence of possible meanings in WordNet (see Fig. 10 for an ontology excerpt).
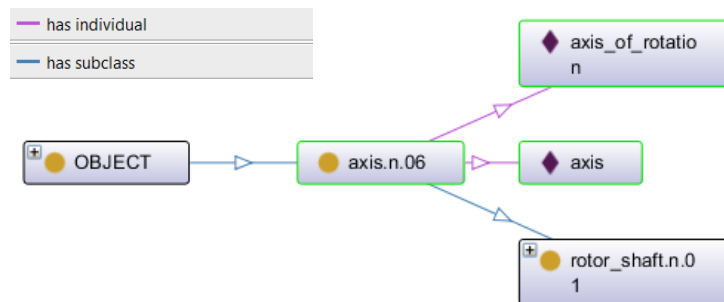
Fig. 10. Example of ontology hierarchical structure and naming principles for object "axis"

While WordNet covers all possible meanings of each word in the English language, the ACAT ontology includes only those senses of verbs and nouns that are typical to robotic actions.

Using the same structural principles and the same naming conventions both in the ACAT ontology and in WordNet allows joint use of information in both ontologies an easy way of supplementing action and object information in the ACAT ontology with structural information from WordNet.

Hypernym/hyponym information for object (or action) names is added to the ACAT ontology in the following way:

1) A feature vector is formed for a selected object (or action) name in the ACAT ontology (context is taken from gloss and examples);
2) All possible senses are extracted from WordNet for the selected object (or action) name;
3) A feature vector in the form of a bag of context words is built for each sense (context is taken from gloss and examples)
4) A word Space Model (WSM), which is based on the hypothesis that words with similar meanings will occur with similar contexts, is used for testing *semantic similarity* of the feature vector for the selected object (or action) name in the ACAT ontology and corresponding feature vectors for WordNet senses.

5) Feature vectors are then compared between each other using the cosine similarity method:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}},$$

where *A* and *B* are the feature vectors of word senses that are being compared.

Cosine similarity ranges from -1 to 1, where -1 means exactly opposite sense, 0 means independence, and 1 shows strong synonyms.

6. Multi-word object name resolution

Object names in the ACAT ontology fall into two groups:

1) Single-word object names;

2) Multi-word object names, usually recognized as collocations.

Our text preprocessing, leading to the extraction of possible action environment objects from corpus texts, involves collocation extraction methods. A collocation is a sequence of words that co-occur more often than it would be by chance (e.g. room temperature).

There are different statistical methods for extracting collocations from the text, such as Mutual Information, the chi-squared test, the Log-likelihood ratio, the Fisher exact test, the Dice coefficient, gravity counts, etc. Experiments showed that for the purpose of identifying action environment elements, log Dice coefficient is adequate:

$$logDice(A, B) = 14 + \log \frac{2|A \cap B|}{|A| + |B|},$$

where $|A \cap B|$ is the frequency of A and B words co-occurrence in text, $|A|$, $|B|$ - frequency of A and B words occurring separately.

Examples of multi-word object names (collocations) in the ACAT ontology are collocations covering domain terms (e.g. *rotor shaft, rotor cap, horizontal surface, measuring beaker*), named entities,

such as chemical elements, names of tools (e.g. *metal spatula*), etc. Multi-word names in the ACAT ontology are positioned (in respect to single words comprising the collocation) using a special relation "collocation".

7. Conclusions and future work

The general structure of the ACAT ontology was designed with the focus on synset hierarchical structure: hypernyms/troponyms for actions and hypernyms/hyponyms for objects (subClassOf property). Another key property of the ontology is the definition of the relations between actions and objects. Specifically, we use restriction properties: with_main_object, with_primary_object with_secondary_object and with_tool to introduce object roles used in the robotic action descriptions into the ontology. With these main points, the ACAT ontology allows essential action environment description by establishing links between an action and a set of environment objects. The object categorization is achieved using VerbNet semantic patterns and WordNet hierarchical structures. The textual ACAT ontology develops the basis for ADT linking to the ontology, which supplements ACAT data structure with subsymbolic information. In the remaining months of the project the ontology will be used in instruction textual compilation for final adjustments of the textual compiler and for measuring of the defined benchmarks.

8. References

1. Markievicz, I., Kapočiūtė-Dzikienė, J., Tamošiūnaitė, M. and Vitkutė-Adžgauskienė, D. (2015) Action Classification in Action Ontology Building Using Robot-Specific Texts, Information Technology and Control, 2015, Vol 44, No.2, 155-164.
2. Markievicz, I, Vitkute-Adzgauskiene, D. and Tamosiunaite, M. (2013) Semi-supervised Learning of Action Ontology from Domain-Specific Corpora. Information and Software Technologies. Springer Berlin Heidelberg, 2013, 173-185
3. Kotsiantis S. B.: Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 2007, 31:249–268.