# SW-CASPAR: Reactive-Cognitive Architecture based on Natural Language Processing for the task of Decision-Making in the Open-World Assumption

Carmelo Fabio Longo[1], Corrado Santoro[1], Domenico Cantone[1], Marianna Nicolosi-Asmundo[1] and Daniele Francesco Santamaria[1]

[1]*Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria, 6, 95125 Catania, Italy*

**Abstract**

This paper addresses the issue of nowadays vocal assistants cognitive lacks, which are able to execute from vocal commands only simple plans without higher capabilities of decision-making. In this work we propose an *open-world assumption* transposition of the cognitive architecture CASPAR, whose heuristic takes into account of meta-reasoning in the *closed-world assumption*, namely SW-CASPAR. Such a cognitive architecture is also provided with a module for semi-automatic *ontology learning* from sentences in natural language, reflecting the domain with an instance of a novel foundational ontology called Linguistic Oriented Davidsonian Ontology (LODO), with the aim of increasing the deepness of reasoning without compromising linguistic-related features. LODO is inspired by the *First-Order Logic Davidsonian* notation and serialized in OWL 2. A case-study applied to automation on health scenarios is also provided.

**Keywords**

Cognitive Architecture, Natural Language Processing, Artificial Intelligence, Semantic Web, Internet of Things, Ontology Learning, Computational Linguistic

## 1. Introduction

In the last years, the market of the Internet of Things (IoT) has become quite disruptive by changing the lifestyle of thousands of people. Thanks to a considerable number of sensors and actuators interconnected with each other, a great number of environments, both home and industrial, have been enriched with every kind of automation, whose diversity is bounded only by the imagination of the designer. In most of such environments, especially homes, vocal assistants, together with the rest of the compatible devices, assume a more and more important role. The two major enterprises in the market of vocal assistants, namely *Amazon* and *Google*, by leveraging their massive dataset for training models in the task of vocal commands recognition, aim more at increasing their product's pervasiveness than respect to improving native reasoning capabilities of the latters. In both cases, each vocal command can be related either to a single pair or to a group of request/operation triggered by specific words. Additional features can be

*WOA 2021: Workshop "From Objects to Agents", September 1–3, 2021, Bologna, Italy*

fabio.longo@unict.it (C. F. Longo); santoro@dmi.unict.it (C. Santoro); domenico.cantone@unict.it (D. Cantone); nicolosi@unict.it (M. Nicolosi-Asmundo); santamaria@dmi.unict.it (D. F. Santamaria)

0000-0002-2536-8659 (C. F. Longo); 0000-0003-1780-5406 (C. Santoro); 0000-0002-1306-1166 (D. Cantone); 0000-0003-4456-5110 (M. Nicolosi-Asmundo); 0000-0002-4273-6521 (D. F. Santamaria)

included by developers with the so-called "Skills" or by exploiting external cloud services like IFTTT [16], but with no enhanced reasoning capabilities.

With "enhanced reasoning capabilities" we intend not only the ability to infer the proper association *command→plan* from utterances, but also to combine facts with rules in order to infer new knowledge and help the user in decision-making tasks. In order to help users in their cognitive processes by using a form of logic reasoning, we must provide the assistant with a combination of facts and rules or, more simply, we must give assistants the capabilities of freely and implicitly extracting what they require from texts in natural language.

In a prior work [1, 2], some of the authors of this paper designed a cognitive architecture called *CASPAR* that instantiates cognitive agents provided with both reactive and cognitive reasoning. Such agents are able to reason in a process subordinated by a further level of reasoning, namely *meta-reasoning*, in a conceptual space, whose content is made of facts and axioms in first-order logic with the closed-world assumption. An important variation of such architecture, which would make it suitable for different scenarios and represents the main motivation of this paper, consists in reasoning over shared *ontologies* in the open-world assumption, in other words by leveraging the *Semantic Web*.

Ontologies are formal, explicit specification of a shared conceptualization [3]. The Semantic Web, with all its layers and frequent updates, can be considered the very backbone of nowadays ontologies.

The closed-world assumption applies when a system has complete information, like many database applications. On the contrary, the open-world assumption applies when a system has incomplete information. For example, consider a patient's clinical history system. If the patient's clinical history does not include a particular allergy, it would be incorrect to state that the patient does not suffer from that allergy; it is unknown if the patient suffers from that allergy, unless more information is given to disprove the assumption.

On the basis of the above, we designed a variation of CASPAR, namely SW-CASPAR[1], capable of parsing IoT commands from natural language, subordinating them by a meta-reasoning on the Semantic Web. An important effort in the development of such architecture has been given by the Owlready [4] library, which gives also the chance of reasoning in the *local closed world*[2], as reported in [5].

SW-CASPAR relies on the support of a module called *Ontology Builder*, which dynamically creates domain-legacy ontologies serialized in OWL 2, in order to allow for meta-reasoning. Nonetheless, it can can be used also as an alternative stand-alone tool for creating ontologies, in substitution of other state-of-the-art tools, for several reasons that will be clarified in later on. A Python prototype implementation of SW-CASPAR is also available as a Github repository.[3]

This paper is structured as follows: Section 2 illustrates the issues of ontology learning from natural language; Section 3 describes the state of the art of related literature; Section 4 shows in detail all the architecture's components and underlying modules, whereas Section 5 illustrates the strategy applied for the task of the Ontology Learning; Section 6 depicts a case-study of an agent working on a health scenario, making usage of both reasoning and meta-reasoning in the

[1] Which stand for: **S**emantic **W**eb-CASPAR.

[2] Owlready reasoning capability in the *local closed world* is limited to a set of individuals and classes.

[3] http://www.github.com/fabiuslongo/sw-caspar

Semantic Web; finally, in Section 7 we draw our conclusions and mention some future work perspectives.

## 2. The issue of Natural Language Ontology

The task of learning an ontology reflecting a domain, by means of a description in natural language of the domain itself, implies several bias that must be considered. In general, ontology learning can be accomplished in three ways: manual learning, by leveraging the effort of experts of the domain; cooperative learning, where most or all the task are supervised by experts; semi-automatic learning, where the ontology construction process is performed automatically with limited intervention by users or experts. It is worth mentioning that full automatic learning by a system is still a significant challenge and it is not likely to be feasible [6]. When sentences are not specifically well-formed, the task of ontology learning from natural language can be quite hard, because of all possible semantic ambiguities of idioms and the arbitrary descriptive nature of the world, which can induce morphologically distinct sequences of words to express the same concept.

In [7], it is reported that "Natural Language Ontology is a branch of both methaphysics and linguistic semantic. It aims to uncover the ontological categories, notions and structures which are implicit in the use of natural language, that is, the ontology that a speaker accepts when using a language". But such an acceptance implies several issues to be addressed by an ontology learning system designer. For instance, for a speaker it would be quite natural and simple to express that a certain object does not exist. In a closed-world assumption we can limit to not assert such a concept, or at least to retract the representation of it from a knowledge base. But how to operate in an open-world assumption, in order to let such information participate in a reasoning process in a human-like fashion? How to keep consistency when an information and its complement are possibly both present in an ontology, as it could happen in texts given by a speaker? In linguistic science, intentional objects as *nonexistent* are considered particularly problematic [8]; for instance, having an ontology A representing a domain of existing entities, and another ontology B representing a description in natural language of the same domain, we cannot definitively say that A and B are equivalent, due to a possible introduction in B of entities which not exist but are functional to the arbitrarily descriptive use of words present in the source text of B.

Let us consider the following two sentences: "*Robert walked down the street*" and "*Robert had a walk down the street*". How can one infer that they express the same concept in a decision process? In the second sentence we are in presence of the so-called *deverbal nominalization* of the verb *walk* versus a noun expressing the action of walking. So we are forced to create somehow a bridge between the two sentences, in order to achieve the same result when both are participating in a reasoning process. Similarly, let us consider also the simple verbal phrase "*the friends are happy*" and the snippet "*happy friends*". In this case, we are in presence of a *deadjectival nominalization*, because the adjective *happy* becomes the object of a copular[4] verb

[4] A copular verb is a special kind of verb used to join an adjective or noun complement to a subject. Common examples are: be (is, am, are, was, were), appear, seem, look, sound, smell, taste, feel, become, and get. A copular verb expresses either that the subject and its complement denote the same thing or that the subject has the property

(be), thus the subject (friends) has the property denoted by the adjective; hence, how to express, by means of an ontology, that the two pieces of information yield the same concept?

The above questions and many other issues related to the natural language ontology are described in [7], which can be a good starting point for filling the gap between a domain and an arbitrary description of it given by a speaker.

In [1, 2], the authors gave an overview of CASPAR, from which SW-CASPAR inherits most of its features, so in the next section we limit ourselves to highlight the state-of-the-art concerning ontology learning, whereas motivations and results concerning the road we chose to follow for the design of this architecture's ontology learning system are postponed until Section 5.

## 3. Related works

The architecture explained in this paper inherits all the *cognitive* IoT features from its predecessor, so the reader is referred to [1, 2] for more details. Hence, this section is focused more on the scope of ontology learning from natural language, which is the main additional contribution of this work.

The disruptive growing of textual data on the web, coupled with an increasing trend to promote the semantic web, has made the automatic ontology construction from texts a very promising research area. However, manual construction of ontologies is time consuming as well as an extremely laborious and costly process. For this reason, several approaches have been designed to automatize the ontology learning from text, each with different levels of human interaction. Such approaches can be divided into two categories: linguistic-based and machine learning approaches. Among the linguistic-based approaches, the authors of [9] use semantic templates and lexico-syntactic patterns such as "NP is type NP" to extract hypernym and meronym relations. But it is well known that these approaches have reasonable precision, though they have a very low recall [6]. In order to achieve terms extraction, [10] leverages POS tagging to assign parts-of-speech to each word and a rule-based sentence parser. However, many words are ambiguous and so this approach will lead to a low accuracy, without a valid disambiguation strategy. Our approach, although similar, makes usage also of a performative disambiguation module, described in details in [2], and extracts also conditional-word based axioms. The authors of [11] make use of a dependency parser to map syntactic dependencies into semantic relations. Such approaches are useful for terms and concepts extraction and also for relations discovery, even though they need to cooperate with other algorithms and/or rules for better performance.

As for machine learning approaches, the system ASIUM [12] adopts agglomerative clustering for taxonomy relations discovery. The axioms only express subsumption relationship (IS-A) between unary predicates and concepts. The system OntoLearn [13] extracts only taxonomic relations, taking into account hypernyms from WordNet. The system HASTI [14] builds automatically ontologies from scratch, using logic-based, linguistic-based and statistical-based approaches. It is one of the few systems that try to learn axioms using inductive logic programming, even though they are very general. Furthermore, such a system has the limitation that

denoted by its complement.

each intermediate node, in the conceptual hierarchy, has at most two children. Worth of mentioning, there is also Text-to-Onto [15], which builds taxonomic and non-taxonomic relations that make use of data mining and natural language processing. Other approaches [16, 17, 18] are also interesting, although either they have limitations on the composition of learned concepts or they generate too many hypotheses, making the involved calculation unmanageable.

Besides a large analysis of state-of-the-art, the authors of [19] discuss on reasons and techniques about the usage of deep neural networks in the Ontology Learning. In these cases, neural networks are often hard to train, although in many cases they give better results by using large, domain-related datasets.

## 4. The Architecture

As pointed out in the introduction, the architecture's name derives directly from its predecessor CASPAR, namely SW-CASPAR, which stands for **S**emantic **W**eb-**C**ognitive **A**rchitecture **S**ystem **P**lanned **a**nd **R**eactive, a name that summarizes its inherited features plus the transposition on the Semantic Web. All interacting components are depicted in Fig. 1, highlighted with distinct colours.

The main component of this architecture, namely the *Reactive Reasoner*, acts as "core router" by delegating operations to other components, and providing all needed functions to make the whole system fully operative.

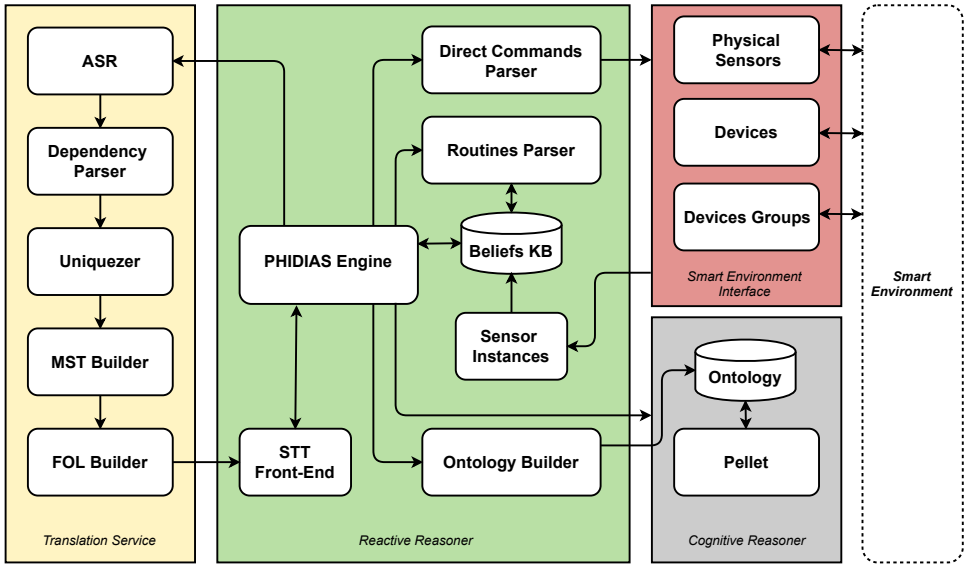


**Figure 1:** The Software Architecture of SW-CASPAR.

This architecture's knowledge base (KB) is divided into two distinct parts operating separately, which we will distinguish as *Beliefs KB* and *Ontology*: the former contains information about physical entities which affect and are affected on the agent, whereas the latter contains conceptual information not directly perceived by agent's sensors, but on which the agent would

make inferences.

The Beliefs KB provides exhaustive cognition about what the agent could expect as input data coming from the outside world; as the name suggests, such a KB is fed by specific beliefs that can - in turn - activate related plans in the agent's behaviour.

The Ontology is defined by triples in OWL 2, and is fed by the Ontology Builder within the Reactive Reasoner.

The two KBs represent, somehow, two different kinds of human being memory: the so-called *procedural memory* or *implicit memory*[20], made of thoughts directly linked to concrete and physical entities, and the *conceptual memory*, based on cognitive processes of comparative evaluation.

As well as in human beings, in this architecture the two KBs can interact with each other in a very reactive decision-making process (meta-reasoning).

### 4.1. The Translation Service

The *Translation Service* component (left box in Figure 1) is a pipeline of five modules with the task of taking a sound stream in natural language and translating it in a *neo-davidsonian* FOL expression inheriting the shape from the event-based formal representation of Davidson [21]. The reader is referred to [1, 2] for a detailed description of such a component.

### 4.2. The Reactive Reasoner

This component (central box in Fig. 1) has the task of letting other modules communicate with each other; it also includes additional modules such as the Speech-To-Text (STT) Front-End, which transforms information coming from other modules in beliefs, IoT Parsers (Direct Command Parser and Routine Parser), Sensor Instances, and Ontology Builder. The Reactive Reasoner contains also the Beliefs KB, which supports both Reactive and Cognitive reasoning.

The core of this component processing is managed by the Belief-Desire-Intention Framework Phidias [22], which gives Python programs the ability to perform logic-based reasoning (in Prolog style) and lets developers write reactive procedures, i.e., pieces of programs that can promptly respond to environment events. For further details about such a component, readers are referred again to [1, 2].

### 4.3. The Smart Environment Interface

This component (upper right box in Fig. 1) provides a bidirectional interaction between the architecture and the outer world. In [23], we have shown the effectiveness of this approach by leveraging the Phidias predecessor Profeta [24], even with a shallower analysis of the semantic dependecies, as well as an operation encoding via WordNet [25] in order to make the operating agent multi-language and multi-synonymous. For more details, the reader is still referred to [1, 2].

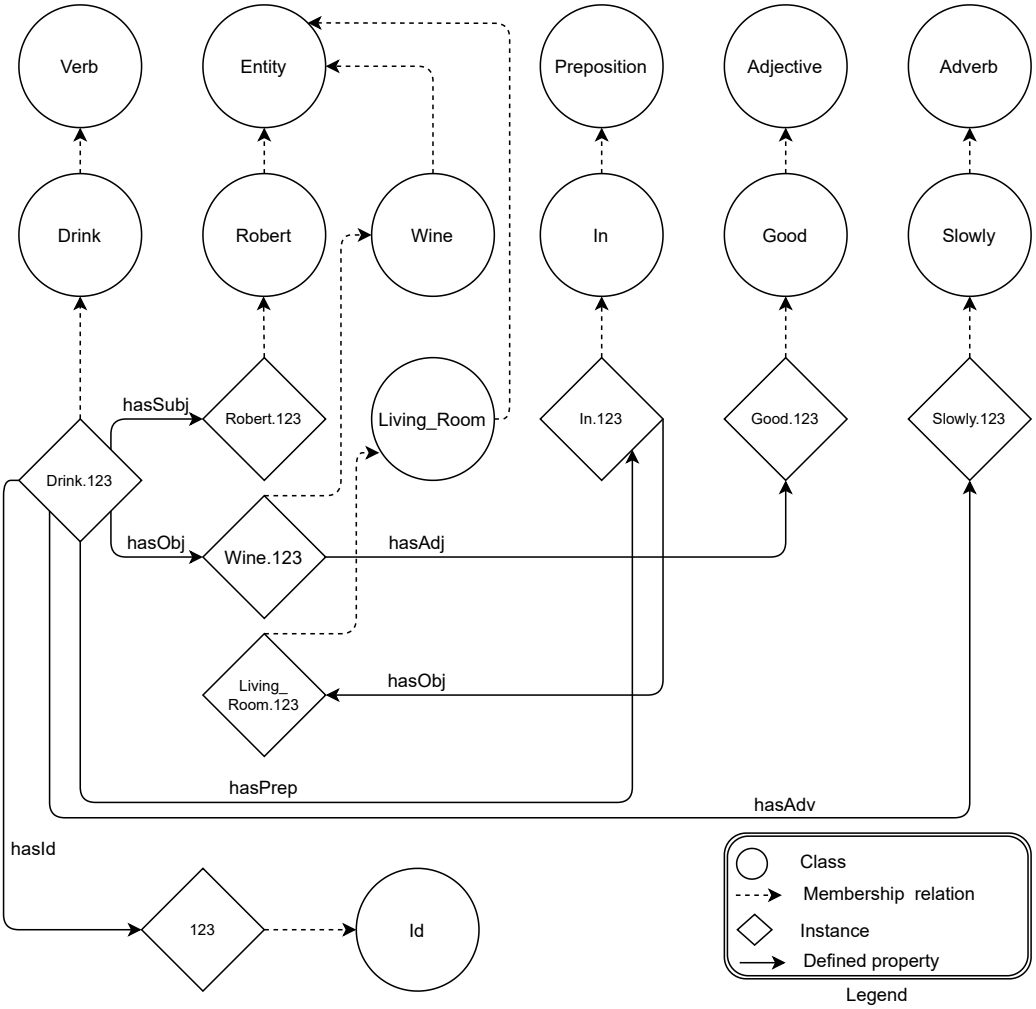


**Figure 2:** A simple instance of LODO ontology related to the sentence (2).

### 4.4. The Cognitive Reasoner

This component (bottom right box in Figure 1) allows an agent to implicitly invoke the Pellet reasoner at runtime, in order to achieve a meta-reasoning subordinating agent's IoT commands. This component comprises the ontology as well, which is fed by the Ontology Builder within the Reactive Reasoner.

## 5. The Ontology Learning

Differently from most approaches to ontology learning, in this paper we give up the idea that whatever such approach will be, it will suffer from the biases related to the ontology of natural language. In light of that, firstly we create a First-Order Logic representation directly linked to

the linguistic features of the sentences in exam, which is provided by the Translation Service component; secondly, whether required, we provide the ontology-specific SWRL [26] rules, whom extend the OWL 2 expressiveness by adding Horn-like axioms. Such rules, as we can see next in the paper, will contribute to fill the gap between the ontology itself and the expected reasoning in human-like fashion. The core module of the Translation Service, as depicted in Fig. 1, is the *Macro Semantic Table* (MST) Builder. Since the MST Builder is made of production rules taking into account relations (dependencies) between words, as long as such relations are treated properly by some rules, the accuracy of the conversion from natural language to logical form can be clearly considered equal to the accuracy of the dependency parser. As for the latter, which in this work's case-study is spaCy [27], in the author's website[5] it is reported to have a state-of-the-art accuracy of 90% for all trained models available for the english idiom.

As reported in detail in [1, 2], the *Translation Service* translates a text in natural language into a First-Order Logic (FOL) expression, inheriting the shape from the event-based formal representation of Davidson [21]. Thus, having such a FOL representation, the module STT Front-End, taking into account the Part-of-Speech which are parts of each label's predicates, will assert specific beliefs triggering the production rules system of the Ontology Builder. The latter has the task of physically creating the OWL 2 domain-legacy ontology containing the triples representing all verbal phrases and their satellite semantic parts (nouns, adjectives, prepositions, and adverbs). For its direct derivation from Davidson notation, in this work we define such a family of ontologies as **LODO** (**L**inguistic **O**riented **D**avidsonian **O**ntology). The latter can be considered a foundational ontology, i.e., a specific type of ontology designed to model high-level and domain independent categories about the real world.

The general schema of LODO is quite straightforward. We define *regular verbal phrase* a set of triples in OWL 2 made by the following classes and their instances:

- *Verb*. Each instance of this class represents what comes as verbal phrase in the Davidsonian notation, from the Translation Service. Each individual has the following object properties: *hasId*, having the values of a unique timestamp; *hasSubject*, representing the verb subject in the domain of *Entity*; *hasObject*, representing the verb object in the domain of either *Entity* or *Verb* (in the case of embedded verbal actions). Another property, namely *isPassive*, possibly indicates whether a verbal action is passive or not.
- *Id*. Each instance of this class represents a unique timestamp related to a verbal actions. It takes the value of the object property *hasId* from some instance of *Verb*. As in temporal logic, such value can be useful to deal with inconsistency cases: the higher is the Id, the more valid is the related instance of *Verb*, even when such an instance has the property *hasAdverb* equal to the value of *Not*[6]; notice that a proper SWRL axiom could be also used to invalidate such obsolete individuals, in order to let them not participate in a reasoning process. Furthermore, by taking into account the Part-of-Speech, it can also introduced an object property *hasTime* of such instance, to express the tenses of the verbal actions (Present, Past Tense, Past Participle, Gerund) that respect the timestamp.
- *Entity*. Each instance of this class represents an entity referenced by the object property either *hasSubject* or *hasObject*. Compound nouns are concatenated in order to form a

[5] https://spacy.io/models/en#en_core_web_lg
[6] Negations are treated as whatever adverb.

single individual.
- *Adjective*. Each instance of this class takes the values of the object property *hasAdj* of some instance of *Entity*.
- *Preposition*. Each instance of this class represents a preposition and it is referenced by the object property *hasPrep* of some instance of either *Verb* or *Entity*. Moreover, each of such instance has the object property *hasObject* referencing some instance of *Entity*.
- *Adverb*. Each instance of this class represents an adverb and has the values of the object property *hasAdv* of some instance of *Verb*.

Together with such taxonomic and non-taxonomic relations, LODO comprises also a group of axioms (or part of them) implicitly created by SW-CASPAR, with the aim of increasing the chances of reasoning/graph matching. Such axioms are summarized as follows:

- *Assignment Rules*. Such rules are implicitly asserted in the presence of a FOL expression representing a copular verb (possibly identified also by its synset) and its satellite predicates. Formally, in the presence of the following Davidsonin FOL expression coming from the Translation Service in Fig. 1:

  $$\text{Subject:POS}(x_1) \wedge \text{Cop:POS}(e_1, x_1, x_2) \wedge \text{Object:POS}(x_2)$$

  where each predicate has its own Part-of-Speech (POS) tag. Such a expression will trigger the assertion of the following SWRL rule[7]:

  $$\textit{Subject(?x) -> Object(?x)} \tag{1}$$

  That's because a copular verb (such as *Be*, for instance) is an intransitive verb but identifying its subject with its object; hence, in this case, the class membership of the verb's object will be inherited by the subject.
- *Legacy Rules*. Such rules are implicitly asserted together with the *Assignment Rules*, to let a copular verb's subject inherit both adjectives and prepositions properties of the verb's object. Formally, considering (1), the corresponding legacy rule will be the following:[8]

  $$\textit{Subject(?x2), Object(?x1), hasAdj(?x1, ?x3), Adjective(?x3) -> hasAdj(?x2, ?x3)}$$

- *Deadjectival Rules* (optional). In the presence of an instance of *Adjective*, such rule asserts a new deadjectivated instance of the latter as new membership of the adjective related noun. Formally:

  $$\textit{Entity(?x1), hasAdj(?x1, ?x2), Adjective(?x2) -> Entity(?x2)}$$

- *Deverbal Rules* (in progress of development). In the presence of an instance of *Verb*, such rules assert a new deverbalized instance of the latter having the same entities as the former.

[7] By omitting the POS, which can be also included in classes labelling.
[8] Similarly for preposition, by changing *hasAdj* with *hasPrep*.

- *Implicative Copular Rules.* Such rules take into account implicative axioms, possibly coming from the Translation Service in FOL Davidsonian notation and containing a single copular verb in the implication's head. They are useful to infer new memberships of the initial sentence subject, which must be present also in the body. The production rule of the Ontology Builder for such rules assertion takes into account the following pattern:

$$\mathsf{Subject}(\mathsf{x}_{body}) \wedge ... \implies \mathsf{Subject}(\mathsf{x}_{subj}) \wedge \mathsf{Object}(\mathsf{x}_{obj}) \wedge \mathsf{Cop}(\mathsf{e}_{cop}, \mathsf{x}_{subj}, \mathsf{x}_{obj})$$

  As shown above, the label Subject must be in both left- and right-hand side of the FOL expression; otherwise, in order to replace possible pronouns that invalidate the pattern, an anaphora resolution pre-processing could be required before the Translation Service pipeline in Fig.1. Cop is the label of a copular verb which will be absorbed, permitting the formal assertion of the following pattern:

$$\textit{Subject(?x}_{obj}\textit{), ... -> Object(?x}_{obj}\textit{)}$$

  Any other implicative FOL expression with non-copular verb in the head will be discarded, due to the non-monotonic features of SWRL.
- *Value Giver Statements* (optional). Such a statement contributes to give a value to a data property *hasValue* related to a specified individual, which is parsed by the Ontology Builder by matching the following pattern of beliefs:

$$\mathsf{GND(FLAT, X, Y), ADJ(FLAT, X, "Equal"), PREP(FLAT, X, "To", S), VALUE(FLAT, S, V)}$$

  The first argument (FLAT) of each belief is for distinguishing non-implicative expressions from implicative ones, and even either right- or left-hand side for the latters. The belief GND is related to a ground term with label Y coming from a FOL expression, which corresponds to a couple of class-individual in the ontology. The beliefs ADJ and PREP specify a lexical content among their arguments, while VALUE specifies the value that must be given to the individual corresponding with label Y. The property *hasValue* might be involved in comparison operations in the composition of a SWRL axiom.
- *Values Comparison Conditional* (optional). Such conditionals are parsed from sentences similarly to the *Value Giver Statement*, but they will take place within the body of Implicative Copular Rules.

For instance, let us consider the following sentence:

*Robert slowly drinks good wine in the living room.* (2)

The Translation Service will give the following FOL expression as result:

$$\mathsf{Robert{:}NNP}(\mathsf{x}_1) \wedge \mathsf{wine{:}NN}(\mathsf{x}_2) \wedge \mathsf{drink{:}VBZ}(\mathsf{e}_1, \mathsf{x}_1, \mathsf{x}_2) \wedge \mathsf{slowly{:}RB}(\mathsf{e}_1) \wedge \mathsf{good{:}JJ}(\mathsf{x}_2) \wedge \mathsf{in{:}IN}(\mathsf{e}_1, \mathsf{x}_3) \wedge \mathsf{living{:}NN}(\mathsf{x}_3) \wedge \mathsf{room{:}NN}(\mathsf{x}_3)$$

Then, on the basis of POS and arguments cardinality, the STT Interface will produce the following set of beliefs:

PREP(FLAT, e1, In, x3), ACTION(ROOT, FLAT, Drink, e1, x1, x2), GND(FLAT, x3, Living), GND(FLAT, x1, Robert), GND(FLAT, x2, Wine), GND(FLAT, x3, Room), ADV(FLAT, e1, Slowly), ADJ(FLAT, x2, Good)

Since implicative copular rules can be applied only one time for sentence, the label ROOT as first argument of the belief ACTION has the aim of distinguishing the main verbal action from possible others in the same sentence. The final step is done by the Ontology Builder (within the Reactive Reasoner in Fig. 1), whose production rules will match in a specified order such beliefs, in order to interface with the Owlready libraries and create the OWL 2 ontology.
Fig. 2 depicts such an ontology: the classes in the upper level (*Verb*, *Entity*, *Preposition*, and *Adverb*) are meant to be subclasses of *Things*; the remaining ones in the circles are subclasses of the former; the diamond shaped boxes are individuals whose label contains also a reference of verbal action's Id (the value *123* is not indicative); the latter is also an individual itself, being instance of the class *Id*.
Due to the presence of the adjective *good* related to the individual *wine*, optionally one might activate the SW-CASPAR deadjectival generation rule; then, invoking an OWL 2 reasoner such as Hermit [28] or Pellet [29], the individual *wine* will achieve the new deadjectivated membership *good*.

## 6. Case-Study

In this section we present a simple case of ontology building and reasoning/meta-reasoning, showing how IoT agents based on SW-CASPAR are able to parse natural language commands and reason about their execution in the open-world assumption. Inspired by [5], where the author shows a use-case consisting in reasoning on drugs contraindications in presence of food intolerances, in this case-study we mainly focus on clinical information about a patient's health disorders and known issues of drugs.

We suppose to provide a hospital with a (semi-)automatized drug distribution system based on natural language recognition, or even to build a robot performing such a task. In this scenario, we suppose to define one or more agents assisting physicians in the decision-making task related with the administration of drugs, on the basis of known drug's issues and clinical picture of patients. In order to address such a task, we extended the Smart Environment Interface of SW-CASPAR with the following set of two production rules given by Phidias,[9] which considers a known issue of the drug *Rinazina* concerning a contraindication for hypertensive patients:

```
+INTENT("Rinazina", T) / eval_sem(T, "Hypertensive")) » [say("Nope. Patient is hypertensive")]   (3)
        +INTENT("Rinazina", T) » [exec_cmd("Rinazina", T), say("execution successful")]          (4)
```

Each production rule which begins with +[10] is expressed in a Prolog-like sintax, where the left hand-side comprises a belief we want the rule to match with (INTENT), conditioned by

[9] For the sake of the case-study we consider a simplified form of rules. The reader id referred to the Github repository for more details.

[10] Thanks to an *ad-hoc* operators override in Python.

other beliefs (in this case the Active Belief `eval_sem`); the right-hand side comprises, in square brackets, the plan to execute when the rule is triggered.
We now generate the domain-legacy ontology exploiting LODO with all the required information about the patient *Robinson Crusoe* and his health disorders starting from the following sentences:

*Robinson Crusoe is a patient*
*Robinson Crusoe has diastolic blood pressure equal to 150*
*When a patient has diastolic blood pressure greater than 140, the patient is hypertensive*

The first sentence is parsed as both regular verbal phrase and assignment rule, whose related classes and individuals are shown in Fig. 3. Classes and instances have similar names (except for the timestamp) by the virtue of *punning patterns*, which increase the chances of reasoning, whereas a unique timestamp is adopted for all the elements of the same verbal phrase.

Together with classes and individuals, the corresponding legacy rules are also asserted (first and third rule of Fig. 4) allowing *Robinson Crusoe* to inherit all the features of the individual *patient* in an analogous way as the speaker's knowledge flow in the scope of the same discourse. The developer might also provide customized IRI,[11] either manually or automatically, by means of a pre-compiled association table.

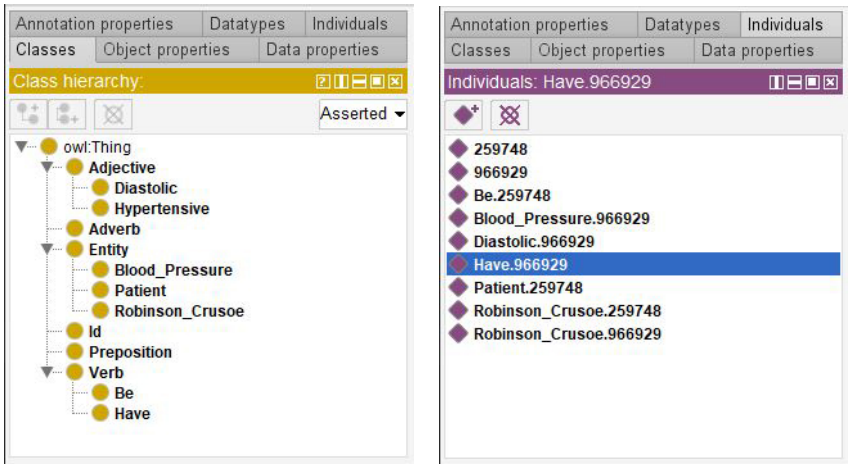


**Figure 3:** The LODO taxonomic relations and instances of the case-study

As shown in Fig. 4, the second sentence is parsed as regular verbal phrase containing also a value giver statement (depicted in Fig. 5).

The third sentence is parsed by the Translation Service as FOL expression containing an implication such as:

$$\text{Have:VBZ}(e_1, x_1, x_2) \wedge \text{Blood:NN}(x_2) \wedge \text{Patient:NN}(x_1) \wedge \text{Pressure:NN}(x_2) \wedge \text{Than:IN}(x_2, x_5) \wedge \text{Diastolic:JJ}(x_2) \wedge \text{Great:JJR}(x_2) \wedge \text{140:CD}(x_5) \Longrightarrow \text{Patient:NN}(x_3) \wedge \text{Hypertensive:JJ}(x_4) \wedge \text{Be:VBZ}(e_2, x_3, x_4)$$

[11] Internationalized Resource Identifier


**Figure 4:** The LODO rules of the case-study

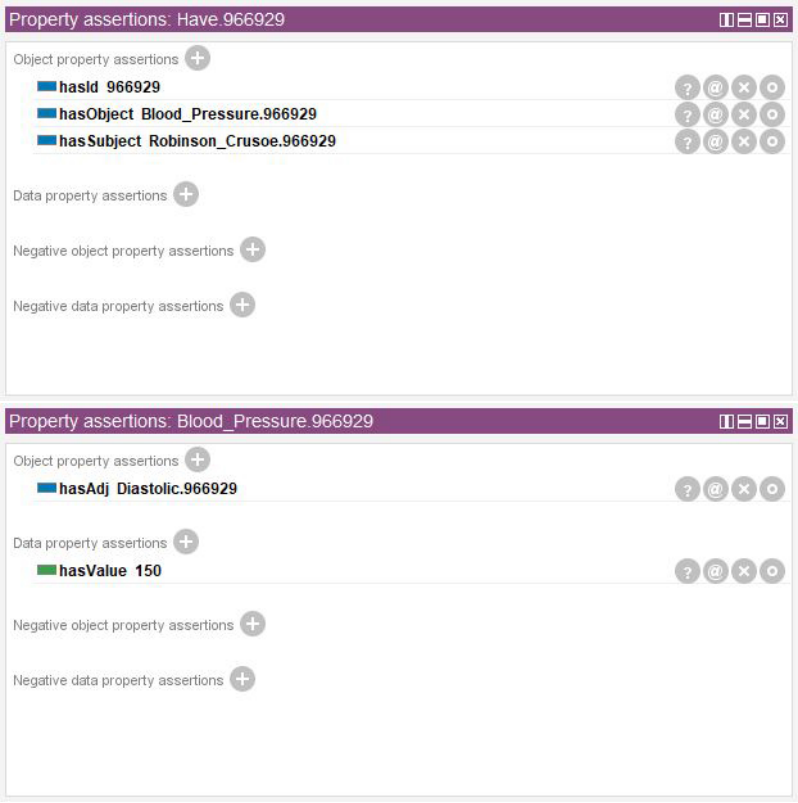


**Figure 5:** The LODO non-taxonomic relations of the case-study

Then the Ontology Builder asserts an implicative copular rule containing a value comparison conditional (the second entry in Fig.4), without creating individuals linked together by the same timestamp.

At the end of the ontology building process, the agent is ready to parse a command containing the following text:


**Figure 6:** Inferred LODO membership after reasoning.

*Give Rinazina to Robinson Crusoe*

As the agent invokes the Pellet reasoner and checks for the membership of *Robinson Crusoe* to the class *Hypertesive* (as well as in Fig. 6 with Protégé), after a successful meta-reasoning of the Active Belief `eval_sem`, the production rule 3 will match with the Beliefs KB content and the command will be discarded with an objection message from the agent. Otherwise, the production rule 4 will match and the command will be executed without any objection message. Of course, the meta-reasoning can involve also more complex queries expressed in SPARQL language and even in local closed world.

## 7. Conclusions and Future Work

In this paper, we presented the design of a cognitive architecture called SW-CASPAR, able to parse IoT commands from natural language and then to execute related plans. Such commands can be subordinated by a further level of (meta-)reasoning in the open-world assumption. Meta-reasoning is achieved by invoking the reasoner over an ontology representing the agent's world serialized in OWL 2, whose semantic is strictly related to the linguistic features of the idiom. The ontology is built by a semi-automatic process, taking into account the issues of natural language ontology, in order to fill the gap between expressiveness and reasoning. All the built ontologies reflect the specifications of the LODO family, which can be considered a foundational ontology aiming both at keeping the expressiveness of Davidsonian notation and at maximizing the chances of successful reasoning.

As future work, we intend to address in more depth other issues and ambiguities of natural language ontologies, in order to include additional rules in LODO, thus leading to a reasoning more human-fashioned.

## References

[1] C. F. Longo, F. Longo, C. Santoro, A reactive cognitive architecture based on natural language processing for the task of decision-making using a rich semantic, in: 21st

Workshop "From Objects to Agents" (WOA 2020), 2020.
[2] C. F. Longo, F. Longo, C. Santoro, Caspar: Towards decision making helpers agents for iot, based on natural language and first order logic reasoning, Engineering Applications of Artificial Intelligence 104 (2021) 104269. URL: https://www.sciencedirect.com/science/article/pii/S0952197621001160. doi:https://doi.org/10.1016/j.engappai.2021.104269.
[3] D. Oberle, N. Guarino, S. Staab, What is an ontology?, Handbook on Ontologies, 2nd edition, Springer, 2009.
[4] L. Jean-Baptiste, Owlready2 0.31, 2021. URL: https://pypi.org/project/Owlready2/.
[5] J.-B. Lamy, Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies, Artificial Intelligence in Medicine 80 (2017) 11–28. URL: https://www.sciencedirect.com/science/article/pii/S0933365717300271. doi:https://doi.org/10.1016/j.artmed.2017.07.002.
[6] A. Browarnik, O. Maimon, Ontology learning from text, in: The First International Conference on Big Data, Small Data, Linked Data and Open Data, 2015.
[7] F. Moltmann, Natural language ontology, 2017. URL: https://oxfordre.com/linguistics/view/10.1093/acrefore/9780199384655.001.0001/acrefore-9780199384655-e-330. doi:10.1093/acrefore/9780199384655.013.330.
[8] P. van Inwagen, Ontology, Identity, and Modality: Essays in Metaphysics, Cambridge University Press, 2001.
[9] M. Finkelstein-Landau, E. Morin, Extracting semantic relationships between terms: supervised vs. unsupervised methods, in: International Workshop on Ontological Engineering on the Global Information Infrastructure, Dagstuhl Castle, Germany, 1999.
[10] A. S., Part-of-speech tagging and partial parsing, Young S., Bloothooft G. (eds) Corpus-Based Methods in Language and Speech Processing. Text, Speech and Language Technology, Springer, Dordrecht vol 2 (1997) 501–518. doi:https://doi.org/10.1007/978-94-017-1183-8_4.
[11] P. Gamallo, M. Gonzalez, A. Agustini, G. Lopes, V. S. de Lima, Mapping syntactic dependencies onto semantic relations, in: Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engineering, 2002.
[12] D. Faure, C. Nedellec, Knowledge acquisition of predicate argument structures from technical texts using machine leraning: The system asium, Knowledge Acquisition, Modelling and Management (1999) 329–334.
[13] R. Navigli, P.Velardi, Learning domain ontologies from document wharehouses and dedicated websites, Computational Linguistics, 30(2), MIT Press (2004) 151–179.
[14] A. A. B. Mehrnoush Shamsfard, Learning ontologies from natural language texts, Int. J. Human-Computer Studies 60 vol 60 (2004) 17–63.
[15] A. Maedche, S.Staab, The text-to-onto ontology learning environment, in: Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures, 2000.
[16] S. Gillani, A. Kő, ProMine: A Text Mining Solution for Concept Extraction and Filtering, Springer International Publishing, Cham, 2016, pp. 59–82. doi:10.1007/978-3-319-28917-5_3.
[17] M. R. U. Hahn, S. Schulz, Medisyndikate–design considerations for an ontology-based

medical text understanding system, in: Proceedings of the AMIA Symposium (p. 330), American Medical Informatics Association, 2000.
[18] K. Z. E. Drymonas, E. Petrakis, Unsupervised ontology acquisition from plain text: The ontogain system, Natural Language Processing and Information System (2010) 277–287.
[19] H. Y. Al-Aswadi Fatima N., Chan, K. H. Gan, Automatic ontology construction from text: a review from shallow to deep learning trend, Artificial Intelligence Review (2020).
[20] D. L. Schacter, Implicit memory: history and current status, Journal of Experimental Psychology: Learning, Memory, and Cognition vol. 13, 1987 (1987) 501–518.
[21] D. Davidson, The logical form of action sentences, in: The logic of decision and action, University of Pittsburg Press, 1967, p. 81–95.
[22] F. D'Urso, C. F. Longo, C. Santoro, Programming intelligent iot systems with a python-based declarative tool, in: The Workshops of the 18th International Conference of the Italian Association for Artificial Intelligence, 2019.
[23] C. F. Longo, C. Santoro, F. F. Santoro, Meaning Extraction in a Domotic Assistant Agent Interacting by means of Natural Language, in: 28th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, 2019.
[24] L. Fichera, F. Messina, G. Pappalardo, C. Santoro, A python framework for programming autonomous robots using a declarative approach, Sci. Comput. Program. 139 (2017) 36–55. URL: https://doi.org/10.1016/j.scico.2017.01.003. doi:10.1016/j.scico.2017.01.003.
[25] G. A. Miller, Wordnet: A lexical database for english, in: Communications of the ACM Vol. 38, No. 11: 39-41, 1995.
[26] World Wide Web Consortium, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004. URL: http://www.w3.org/Submission/SWRL/.
[27] H. Matthew, spacy: Industrial-strength natural language processing, https://spacy.io, 2017.
[28] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, Z. Wang, HermiT: An OWL 2 Reasoner, Journal of Automated Reasoning 53 (2014) 245–269.
[29] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, Y. Katz, Pellet: A practical OWL-DL reasoner, Web Semantics 5 (2007) 51–53.