

Supporting the Creation of Some Class of Well-founded OWL-DL Ontologies

Jolanta Cybulka

*Institute of Control and Information Engineering
Poznań University of Technology, Poznań, Poland
E-mail: jolanta.cybulka@put.poznan.pl*

Received: 09 December 2016; revised: 24 February 2017; accepted: 24 February 2017; published online: 18 March 2017

Abstract: We present the method for creating semantically rich OWL-DL ontologies. Such ontologies contain a top part consisting of the most general concepts that are linked by carefully designed basic formal relations, which is why the ontologies are called *well-founded*. As the top part we apply the foundational paradigm of *constructive descriptions and situations* c.DnS known from literature, using it to obtain the method of a layered ontology creation. Every layer may contain ontological *capsules*, i.e. modules that are defined according to an expressive content pattern (formed on the basis of c.DnS), which makes the whole ontology well-founded. The method is equipped with the semantically strong computational tool that effectively supports a user in the process of creation and population of the considered well-founded ontologies.

Key words: conceptual modelling, well-founded ontology, ontology development tool, foundational ontology

I. INTRODUCTION

Nowadays, ontologies are often used both as reference conceptual models of a domain and as the explicit representation of semantics exploited in information systems. Usually, the first type ontologies are more (ontologically) expressive than the second, but more and more frequently also the latter become complex and semantically rich. In the paper we promote building such ontologies to be used as semantic resources in information systems. Expressive ontologies of this kind are often called *well-founded* because they contain the top part consisting of the most general concepts that are linked by carefully designed basic formal relations – as such, well-founded ontologies represent philosophically and linguistically grounded view on reality.

There exist many methodologies of ontology creation equipped with supporting tools. In papers [9, 10] one may find still cited basic overviews of ontology creation methodologies that were proposed by large teams of researchers. The newer overview in [15] states (mainly on the basis of [16]) that there exist 20 documented ontology engineering methodologies. Authors of all overviews propose generic criteria based on which the methodologies are analyzed and clas-

sified. For example, the criteria concern the strategies for building ontologies (i.e. application-dependent, application-independent or application semi-dependent), the existence of recommended modelling processes (similar to the ones used in software engineering, such as waterfall, iterative or agile development models), the possibility of collaborative construction of an ontology, the tool-support and the specification of the primary target group of users of the methodology (either ontology engineers or domain experts). The best known, analyzed and classified methodologies are: TOVE, Enterprise Ontology, KACTUS, SENSUS, METHONTOLOGY, On-To-Knowledge, DILIGENT, DOGMA, UPON, HCOME or the proposal of eXtreme ontology Design (XD) relying on pattern-based ontology design technique. Using the ontology design patterns forms a part of a framework that was formalized as an OWL-DL ontology named C-ODO (Collaborative Ontology-Design Ontology) [14], which was developed within the EU NeOn (Networked Ontologies) project [17]. The XD methodology is a little bit more elaborated in section 2.

In the paper we are concerned with the sub-problem of ontology creation, namely with designing of the tool that supports the user in manual creation of a certain class of well-founded ontologies.

The **contributions of the presented work** are listed below.

1. We propose the c.DnSPL¹ method for creating semantically expressive ontologies, which strongly relies on the foundational ontology of *constructive descriptions and situations* elaborated in [1] (c.DnS). In the c.DnSPL method the modeler looks at the modelled reality through the expressive content pattern formed on the basis of the c.DnS foundational ontology. This distinguishes the ontology created in such way from simple ontologies which have poorer semantic assumptions. Using the considered content pattern, the modeler builds an ontological *capsule* that forms a module of the created ontology. This module groups together concepts concerned with some facet of a modelled domain. Such capsules can be specialized by subsumption and interlinked by other ontological relations, forming a capsular (modular) structure. This structure reflects the structure of a modelled reality where some domain may have subdomains and elements of a subdomain may be related to elements of some other subdomains. Additionally, subsumption determines layering of capsules of an ontology – this feature is more elaborated in section 2.
2. The method is equipped with a tool which is a kind of a supporting assistant. The tool has several options. The most basic one is the generation, in dialogue with the user, of a new capsule (in OWL-DL [5]) of a c.DnSPL well-founded ontology. The option is used when the user wants to model some new domain of interest from scratch. There is also a possibility to specialize the already existing capsule, meaning that the new specializing capsule is generated (in dialogue with the user, too) to model from scratch a subdomain of some already modelled domain. Sometimes there already exists an ontology of a domain of interest but its concepts are semantically underspecified. For example, such an ontology may encode in OWL (or RDFS) some database schema and may contain only lower-level concepts, whose intended semantics is only superficially specified by arbitrarily chosen relations. Even so such an ontology may be useful for some purposes, but it may also serve as a primary source of domain concepts to which a deeper semantics should be assigned. We call the process of assigning a new deep semantics to concepts (this is done by expressing them with the use of basic relations of the foundational content pattern) of an existing ontology by the term *reification* (it is one more option of the tool). To reify any arbitrary ontology with the use of the supporting tool one has to first generate a capsule specifying the semantics of a domain, then read-in an existing ontology

and assign the new semantics to the chosen concepts (the old semantics of them is also preserved). All the obtained c.DnSPL ontologies may be further manually processed with the use of, for example, Protégé (<http://protege.stanford.edu/>) ontology editor.

The created semantically rich ontologies may be used for different purposes – for example, suppose that we created an ontological capsule representing the semantics of events in some domain of interest, because we want to automatically extract information on such events from natural language texts. The ontology may serve as a basis of the process of semantics-driven generation of extraction templates, which is described in [12]. The usage of such generated templates is shown in [13].

The paper is structured as follows. In section II the c.DnSPL ontology is defined. At first, the idea and the informal specification of its main features are given, in relation to the XD pattern-based ontology design technique. Then we formalize the (terminological part of the) ontology in the style of P. Cimiano [4]. At the end of the section the algorithm of a c.DnSPL capsule generation is presented, which is implemented as the core part of the supporting tool.

In section III we present the supporting tool in action. First, the process of the user-assisted generation of a capsule of the ontology is illustrated. As an example of the modelled domain of interest a situation of travelling was chosen. Second, the procedure of reifying of an arbitrary travelling ontology into the c.DnSPL paradigm is depicted. In the end, section IV contains concluding remarks.

II. THE WELL-FOUNDED C.DNSPL ONTOLOGY

We consider the creation of conceptual models in the form of an ontology that has a layered structure. Ontological layers are organized by the subsumption relation. Each layer contains one or several elements (called ontological *capsules*), each of which is constructed according to the content pattern which is the generic relation of *constructive descriptions and situations* c.DnS of [1]. One generic capsule (formally, it is a c. DnS relation) is always on top of the ontological structure and forms the foundational layer of any constructed ontology. The pattern-based approach to ontology creation is the kernel of the XD method (eXtreme ontology Design, [3]). It postulates to perceive the modelled domain through several types of patterns, especially a collection of content patterns. Each such pattern is equipped with a set of competency questions representing use cases of a knowledge base that is semantically represented by an ontology. The ontology engineer uses questions to choose a content pattern (an exact, an approximate or a redundant one), which is a small structure and has limited semantic expressiveness. Thus, XD may be considered as a kind of a “middle-out” method of

¹ The name ‘c.DnSPL’ is created after the name ‘c.DnS’ by adding the ‘PL’ suffix that denotes ‘Polish’, meaning that both the method is developed in Poland and entities of the ontology are primary lexicalized in Polish (although the creation of multilingual versions of ontologies is also possible and supported).

ontology creation with the use of the library of elementary content patterns accompanied by operations that can be done on them (such as importing, cloning, specializing, generalizing, composing and extending). In the paper we propose to use one content pattern (a generic $c.DnS$ relation) which is expressive enough to model different domains. In many cases it may be semantically redundant, but this is not a serious problem. In the consecutive layers of the created ontology, this pattern is specialized and the elements (capsules) obtained in this way can be additionally linked by other formal ontological relations. So, we create the ontology in a “top-down” manner. The structure may be represented as a graph with capsules as nodes and ontological relations as vertices. Fig. 1 depicts the exemplary layered ontological structure. Thick arrows represent subsumption while thin ones represent other formal relations. The w_0 layer contains only a generic $c.DnS$ capsule (we also call it a $c.DnSPL_0$, see the equation (2) and the OWL-DL implementation of this generic capsule in [2]). The w_1 layer contains k capsules that specialize the foundational one. In w_2 the capsule $c.DnS_1$ is specialized by m capsules while the capsule $c.DnS_k$ is specialized only by one $c.DnS_{k1}$ capsule. Additionally, some elements of $c.DnS_{1m}$ capsule are linked by other relations with elements of $c.DnS_{11}$ (in the same layer) and $c.DnS_k$ (from the former layer). The main advantage of such an approach is the logical (as opposed to physical) grouping (encapsulation) of concepts concerned with some facet of a modelled domain. They are seen “at once” while being in fact nested in a dense net of concepts of the whole ontology.

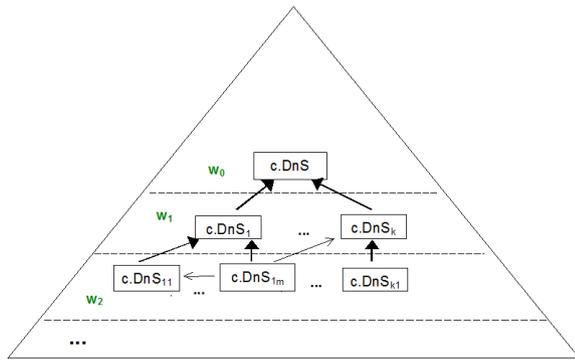


Fig. 1. The layered and capsular tree-like structure of an ontology

More formally, after [4], an ontology O (1) is a structure of the form:

$$O = \langle C, \leq_C, R, \sigma_R, \leq_R, A, \leq_A, \sigma_A, T \rangle. \quad (1)$$

1. C is a set of names of concepts ordered by the \leq_C partial order relation; the system $\langle C, \leq_C \rangle$ forms an upper semi-lattice with the maximal element T that is the most general concept

2. function $\sigma_R : R \rightarrow C+$ is a relation signature that assigns a $C+$ word to the relation name (sign + denotes the positive closure of the concatenation of words), which denotes the arity and typing of arguments of the relation
3. R is a set of names of relations ordered by \leq_R partial order (every relation of a pair of relations in this connection has the same arity and the types of arguments of the first relation precede by means of \leq_C their counterparts in the second relation)
4. function $\sigma_A : A \rightarrow C \times T$ is a signature of attributes that assigns to the name of an attribute a pair: a name of a concept and a data type
5. A is a set of names of attributes ordered by \leq_A partial order (the type of the argument of the first attribute precedes the type of the argument of the second attribute and the datatypes are the same)
6. T is a set of data types.

The presented definition covers only one facet of the ontology, namely its main terminological properties. Thus it may be further specified to contain: more detailed properties of concepts (in the form of axioms), the assertions concerning examples (instantiations) of concepts, relations and attributes, and lexicons defining signs attached to concepts, relations, attributes and their examples. The presented definition is sufficient in our considerations.

Fig. 1 informally depicts the terminological structure of the $c.DnSPL$ ontology. It contains layers that include 9-argument relations (we call them capsules). The set of these relations is partially ordered and the ordering forms the upper semi-lattice of relations with the maximal element $c.DnSPL_0$ (which is the only element of the foundational level of the ontology). The figure illustrates a tree-like structure of connections (expressed by subsumption) between relations but in general, due to multi-inheritance it may be a graph.

Let us now concentrate on the foundational level relation $c.DnSPL_0$ with the signature $\sigma_{Rc.DnSPL}$:

$$\sigma_{c.DnSPL}(c.DnSPL_0) = \langle \text{Description, Situation, Concept, GroundEntity, SocialAgent, AgentCollective, InformationObject, TimeInterval, TimeInterval} \rangle \quad (2)$$

Sort names in the signature represent ontological concepts that are linked via binary formal relations (below, in the paraphrase, they are written in italics). Using concept names and relation names the following textual paraphrase, which explains the essence of the modeling of the reality, can be given:

“A *SocialAgent* perceives the modelled state-of-affairs in (the first) *TimeInterval*. This state-of-affairs is ontologically reified as *Situation* that is *set-for* by *GroundEntities*. A *SocialAgent re-describes* in (the second) *TimeInterval* a *Situation* in the form of a *Description*. A *SocialAgent classifies* *GroundEntities* via *Concepts* that are *defined* or *used* in a *De-*

scription. Every Situation *satisfies* some Description which is *expressed* by some InformationObject. A SocialAgent is a *member of* an AgentCollective that *share* or *assume* a Description.”

The $c.DnSPL$ ontology is defined as (3) (we ignore lexicalizations of concepts and relations):

$$\begin{aligned} O_{c.DnSPL} &= \\ &= \langle C, \leq_C, R2 \cup R3 \cup \{\text{redescription4}\} \cup R_{c.DnSPL}, \\ &\leq_{R2}, \leq_{R3}, \leq_{R_{c.DnSPL}}, \sigma_{R2}, \sigma_{R3}, \sigma_{\text{redescription}}, \sigma_{R_{c.DnSPL}} \rangle \end{aligned} \quad (3)$$

C is the set of concepts names, including those given in (2). Concepts are ordered by a partial order relation \leq_C with Entity as the most general concept. The set $R2$ represents binary relations (they have inverses) with signatures $\sigma_{R2}: R2 \rightarrow C \times C$, for example:

$$\begin{aligned} \sigma_{R2}(\text{defines}) &= \langle \text{Description}, \text{Concept} \rangle \\ \sigma_{R2}(\text{satisfies}) &= \langle \text{Situation}, \text{Description} \rangle. \end{aligned} \quad (4)$$

Binary relations are ordered by subsumption \leq_{R2} . Ternary relations $R3$ have a time interval as an argument $\sigma_{R3}: R3 \rightarrow C \times C \times \{\text{TimeInterval}\}$ (their inverses ignore the time argument). For example:

$$\begin{aligned} \sigma_{R3}(\text{assumes}) &= \langle \text{SocialAgent}, \text{Description}, \text{TimeInterval} \rangle \\ \sigma_{R3}(\text{classifies}) &= \langle \text{Concept}, \text{Entity}, \text{TimeInterval} \rangle. \end{aligned} \quad (5)$$

Ternary relations are ordered by \leq_{R3} . The redescription4 relation distinguishes a GroundEntity that is re-described by a SocialAgent on the basis of the observation in TimeInterval of a state-of-affairs that is reified in the ontology as a Situation:

$$\begin{aligned} \sigma_{\text{redescription}}(\text{redescription4}) &= \\ &= \langle \text{SocialAgent}, \text{GroundEntity}, \text{Situation}, \text{TimeInterval} \rangle. \end{aligned} \quad (6)$$

The $R_{c.DnSPL}$ contains the foundational relation $c.DnSPL_0$ (2) and other 9-ary relations that have the signature (7).

$$\begin{aligned} \sigma_{R_{c.DnSPL}}: R_{c.DnSPL} &\rightarrow \\ &\rightarrow C_1 \times C_2 \times C_3 \times C_4 \times C_5 \times C_6 \times C_7 \times C_8 \times C_9, \end{aligned} \quad (7)$$

where

$$\begin{aligned} C_1 &\subset C \wedge \forall c \in C_1 c \leq_C \text{Description}, \\ C_2 &\subset C \wedge \forall c \in C_2 c \leq_C \text{Situation}, \\ C_3 &\subset C \wedge \forall c \in C_3 c \leq_C \text{Concept}, \\ C_4 &\subset C \wedge \forall c \in C_4 c \leq_C \text{GroundEntity}, \\ C_5 &\subset C \wedge \forall c \in C_5 c \leq_C \text{SocialAgent}, \\ C_7 &\subset C \wedge \forall c \in C_7 c \leq_C \text{InformationObject}, \\ C_6 &\subset C \wedge \forall c \in C_6 c \leq_C \text{AgentCollective}, \\ C_8 &\subset C \wedge \forall c \in C_8 c \leq_C \text{TimeInterval}, \\ C_9 &\subset C \wedge \forall c \in C_9 c \leq_C \text{TimeInterval}. \end{aligned}$$

² The abbreviation ns is created from the first letters of a name of a situation.

The $R_{c.DnSPL}$ set is ordered by $\leq_{R_{c.DnSPL}}$ relation that is a reflexive and transitive closure of a direct preceding relation $\prec_{R_{c.DnSPL}}$ (10). Let $c.DnSPL_m$ and $c.DnSPL_n$ be two relations (8) and (9) (with variables x_i and y_i , $i \in [1, 9]$) that belong to $R_{c.DnSPL}$, with the following signatures:

$$\begin{aligned} \sigma_{c.DnSPL}(c.DnSPL_m) &= \langle x1 \leq_C \text{Description}, \\ &x2 \leq_C \text{Situation}, x3 \leq_C \text{Concept}, \\ &x4 \leq_C \text{Entity}, x5 \leq_C \text{SocialAgent}, \\ &x6 \leq_C \text{AgentCollective}, x7 \leq_C \text{InformationObject}, \\ &x8 \leq_C \text{TimeInterval}, x9 \leq_C \text{TimeInterval} \rangle \end{aligned} \quad (8)$$

$$\begin{aligned} \sigma_{c.DnSPL}(c.DnSPL_n) &= \langle y1 \leq_C \text{Description}, \\ &y2 \leq_C \text{Situation}, y3 \leq_C \text{Concept}, \\ &y4 \leq_C \text{Entity}, y5 \leq_C \text{SocialAgent}, \\ &y6 \leq_C \text{AgentCollective}, y7 \leq_C \text{InformationObject}, \\ &y8 \leq_C \text{TimeInterval}, y9 \leq_C \text{TimeInterval} \rangle \end{aligned} \quad (9)$$

then

$$\begin{aligned} c.DnSPL_m \prec_{R_{c.DnSPL}} c.DnSPL_n &\Leftrightarrow \exists(i \in [1, 2, \dots, 9]) \\ &x_i \prec_C y_i \wedge \forall(j \neq i, j \in [1, 2, \dots, 9])(y_j \prec_C x_j). \end{aligned} \quad (10)$$

The terminological facet of an ontology (3) is augmented with axioms given in [1] (there is no room to present them here).

Now we are ready to sketch the algorithm of the creation of a $c.DnSPL$ relation (a capsule of a well-founded $c.DnSPL$ ontology) for any domain of interest. There are two modes of the user-assisted generation of a capsule: a) by the specialization of the foundational generic relation and b) by the specialization of any existing relation. The case a) is representative thus the algorithm is based on it. It is assumed that the version of the $c.DnSPL$ ontology which is encoded in OWL-DL [2] is considered here. In the presented algorithm, to link concepts, say A and B , by the role (description logic relation) r , the subsumption (\Rightarrow) axiom is used with the existential quantifier: $A \Rightarrow \exists r.B$ (A contains the instances, each of which is connected via r with existing instances of B).

Generating a capsule of the well-founded $c.DnSPL$ ontology

Input: A generic $c.DnSPL$ capsule written in OWL, concepts' lexicalizations supplied by the user

Output: An OWL-DL ontology augmented with the generated capsule

Method:

Do the following steps (lexicalizations given by the user are in italics):

1. read-in a *SituationName*, create a concept $cDnSPL_SituationOfSituationName$

2. create the concept *Element-cDnSPL-SituationOfSituationName*
3. create 9 concepts that specialize *Element-cDnSPL-SituationOfSituationName*, consecutively: S_{ns}^2 , D_{ns} , C_{ns} , E_{ns} , A_{ns} , K_{ns} , I_{nz} , T_{S-ns} , T_{D-ns}
4. link the concepts created in 3 by formal ontological relations (absorb them from the generic pattern, i.e. $R2$ and $R3$ from the equation (3)):
 - (a) link the concept *cDnSPL-SituationOfSituationName* to the concept *Element-cDnSPL-SituationOfSituationName* by the *hasProperPart* relation
 - (b) link the concept *Element-cDnSPL-SituationOfSituationName* to the concept *cDnSPL-SituationOfSituationName* by the *isProperPartOf* relation
 - (c) link the concept *cDnSPL-SituationOfSituationName*, consecutively, to the concepts: S_{ns} , D_{ns} , C_{ns} , E_{ns} , A_{ns} , K_{ns} , I_{nz} , T_{S-ns} , T_{D-ns} by the *hasProperPart* relation
 - (d) link E_{ns} to C_{ns} by *isClassifiedBy* and link E_{ns} to S_{ns} by *setsFor*
 - (e) link S_{ns} to E_{ns} by *isSettingFor* and link S_{ns} to D_{ns} by *satisfies* relation
 - (f) link D_{ns} to A_{ns} by *isSharedBy*, link D_{ns} to S_{ns} by *isSatisfiedBy* and link D_{ns} to I_{nz} by *isExpressedBy*
 - (g) link I_{nz} to D_{ns} by *expresses* relation
 - (h) link A_{ns} to the concept *PhysicalAgent* by *isActedBy*, link A_{ns} to K_{ns} by *isMemberOf* and link A_{ns} to D_{ns} by *shares* relation
 - (i) link K_{ns} to A_{ns} by *hasMember*
 - (j) link C_{ns} to E_{ns} by *classifies* relation.
5. Ask the user if he/she wants to continue. If not, save the file and quit, else:
 - (a) read-in descriptions of a situation in the form of a *PlanName* and *GoalOfAPlanName*, specialize D_{ns} by the created concepts *PlanOf PlanName* and *GoalOf GoalOfAPlanName*, link *GoalOf GoalOfAPlanName* to *PlanOfPlanName* by *isMainGoalOf* and link *GoalOf GoalOfAPlanName* to *PlanOf PlanName* by *hasMainGoal* relation
 - (b) ask the user if he/she wants to add classifications of perdurants (entities that happen in time and space) of a situation. If not, go to 5., else read-in: either a task (classifying an action) or an activity description or an event description or a state description – all the types of perdurants are created in the same way, thus we consider a task concept creation only:
 - i. read-in a *TaskName* and specialize the C_{ns} by the created *TaskTaskName* concept
 - ii. ask the user if he/she wants to add classifi-

cations of endurants (persistent entities that are arguments) of the defined task, which are called roles. If not, go to 5b), else iteratively list the types of roles to the user and for the chosen *RoleRoleType* read-in its *RoleName*, specialize C_{ns} by the created *RoleRoleTypeRoleName*.

6. Go to 5.

III. GUIDED CREATION OF A WELL-FOUNDED ONTOLOGY – A SUPPORTING TOOL AND AN EXAMPLE

The algorithm described in section 2 is implemented in Java [5]. As it was mentioned in point 2 of the contributions of the presented paper in the introduction, besides the user-assisted generation of an ontological capsule the tool supports the reification of an arbitrary ontology into a c.DnSPL paradigm. It means that concepts of this ontology are expressed with the use of terms of c.DnSPL well-founded ontology. Let us assume that we are interested in modelling the travelling domain by means of semantically expressive ontology. At first we search the Internet to find any existing travelling ontology written in OWL – we are lucky finding the file *travel.owl* [7], with the taxonomy of concepts depicted in Protégé window in Fig. 2. After analyzing concepts' definitions from the c.DnSPL contents pattern we can conclude that *Destination* represents some description of a travelling situation (a plan of it with a goal), *Activity* is a perdurant within this situation and *Accommodation* plays a locative role of a perdurant. We can interpret *AccommodationRating* as a qualitative parameter of a travelling situation.

However, our goal is to model the considered domain using the expressive view on reality wired in the c.DnSPL ontology. Thus we are to generate a capsule of a travelling situation, with its description in the form of a plan of travelling which has some goal of travelling. The plan uses activity descriptions in which certain entities play certain roles. Let us use the supporting tool (up to now the interface is in Polish, [11]) to generate the skeletal capsule that models a travelling situation, which specializes the generic foundational capsule. The user may choose one of three options (radio buttons):

1. building a new capsule from scratch
2. specializing any existing capsule and
3. finishing the earlier suspended creation of some capsule.

We choose the first option. Then we are to choose the form of the foundational capsule: either

1. taken from system's repository or
2. read-in from the external file.

In the next steps the system interacts with the user according to the algorithm given in section 2, that is the situation name is given by the user and the concepts *cDnSPL-SituationOfTravelling* and *Element-cDnSPL-SituationOfTravelling* are created. After that, the skeleton

of the travelling situation capsule is generated and the user can give names of the plan of the situation and its goal used to create concepts specializing situation's description (*D-SituationOfTravelling*).

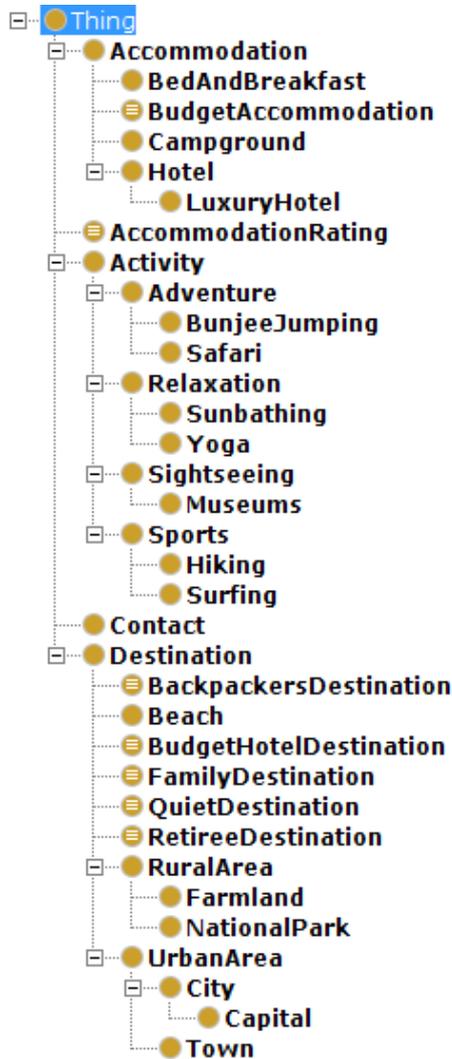


Fig. 2. The taxonomy of concepts of an arbitrary travelling ontology

Consecutively, the user can add names to create descriptions of perdurants and their roles which are used in the plan and specialize the concept *C-SituationOfTravelling*. The result of this part of ontology generation is depicted in Fig. 3 (with the use of Protégé that enables rendering entities by `rdf:label` and `xml:lang="en"`).

After the well-founded ontology of travelling is created (it contains two capsules: the foundational and the generated one) we can reuse the ontology [7] by populating the newly generated capsule with its concepts. Using our tool concepts given in `travel.owl` are subsumed by *Ground entity* of the foundational capsule and then the user can transfer them under *E-SituationOfTravelling* concept that sets up the travelling

situation (Fig. 4).

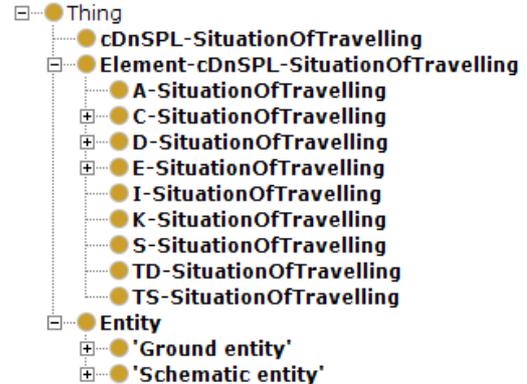


Fig. 3. The taxonomy of concepts of the generated capsule

In our new ontology these 'real' concepts from [7] have counterparts that classify them, giving them new well-founded semantics. They are automatically generated and placed under *C-SituationOfTravelling*. After that the generation of the ontology ends and its results are stored in the `travel-skeleton.owl` file in [8]. The ontology may be further manually refined with the help of Protégé, as for example in the `travel-refined.owl` file in [8]. The taxonomy of concepts of the part of the refined version of the generated ontology is shown in Fig. 5. It can be seen that *Activity-sp* (classifying the *Activity*) is a description of an activity, *Destination-sp* (classifying *Destination*) is a plan of a travelling situation and *Accommodation-sp* (classifying *Accommodation*) is a locative role of an activity. All the concepts are linked by basic formal ontological relations (from the foundational level) gaining new, well-founded semantics. However, the original semantics of concepts (that are subsumed by *E-SituationOfTravelling*) of the reified ontology is still preserved.

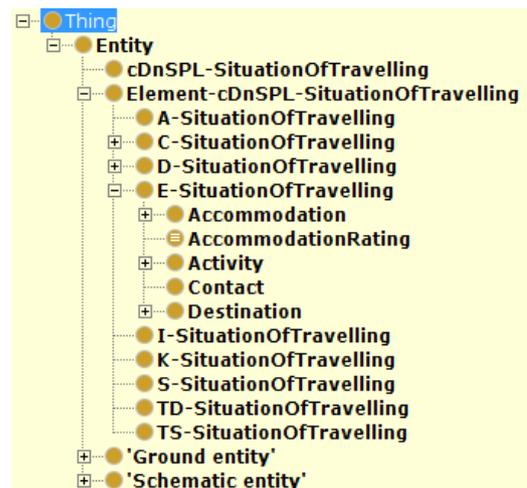


Fig. 4. The taxonomy of concepts after transferring under *E-SituationOfTravelling* and reasoning with FaCT++

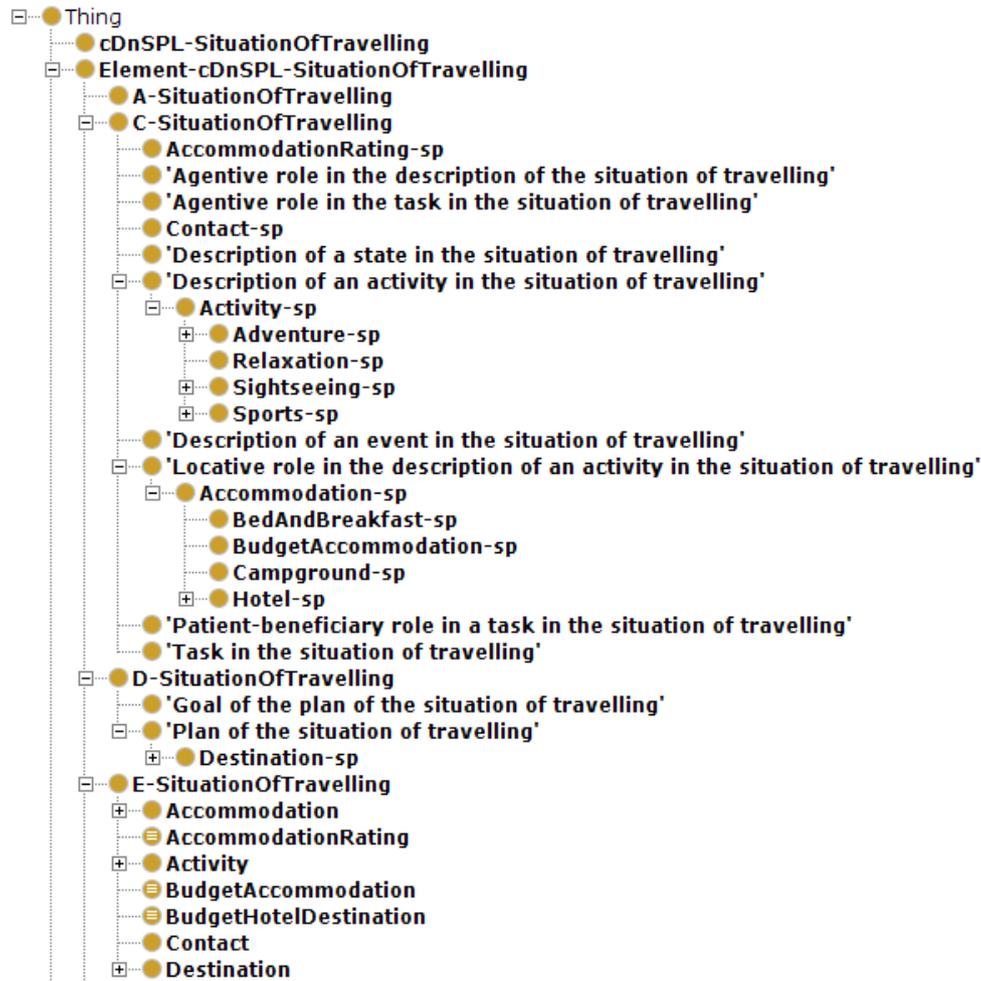


Fig. 5. Part of the taxonomy of concepts of the refined version of the generated capsule

IV. CONCLUSIONS

The main goal of our research is to develop a method for building semantically expressive (well-founded) ontologies. As it was said earlier, such ontologies contain the top part consisting of the most general concepts that are linked by carefully designed basic formal relations. There exist several proposals of such top (also called foundational) ontologies – developing our method we highly benefited from the foundational ontology of *constructive descriptions and situations* c.DnS described in [1]. Creating well-founded ontologies is not an easy task, thus the method should be equipped with the supporting tool. We implemented a software assistant that enables the interactive generation of well-founded c.DnSPL ontologies. The tool guides a user through the ontology creation process, as a result of which the capsule of the ontology in OWL-DL is generated. The obtained capsule may be additionally populated with concepts of any existing ontology that concerns the ontologically modeled domain. This reused ontology concepts gain new semantics while their original one is still preserved. The tool is implemented in Java and

it is described in details in the BA thesis in Engineering [6]. Although the experiments done (qualitative testing) with the system are promising, the tool is still in an alpha version [11] (it is not free from errors) and needs improvement. The next step is to implement an English version of the interface of the tool. Also, the processes of ontology capsule generation and that of the capsule refinement by means of arbitrary ontology reification should be both improved to avoid some manual work that is now needed to be done on the resulting .owl file. After these refinements the tool should be tested through the generation of large-scale ontologies and eventually the system may be quantitatively evaluated.

Acknowledgement

The author wishes to thank Aleksandra Piotrowicz and Izabela Procek who are the developers and maintainers of the tool.

References

- [1] A. Gangemi, J. Lehmann, C. Catenacci, *Norms and plans as unification criteria for social collectives*. In: *Proceedings of Dagstuhl Seminar 07122, Normative Multi-agent Systems*, G. Boella, L. v. d. Torre, H. Veerhagen (Eds.), Vol. II ISSN 1862-4405, pp. 48-87, <http://drops.dagstuhl.de/opus/volltexte/2007/910>, 2007 [retrieved: January 2017].
- [2] *OWL-DL implementation of the generic relation of c.DnSPL ontology*, <http://users.man.poznan.pl/jolac/cDnSPL/c.DnSPL.zip>, [retrieved: January 2017].
- [3] A. Gangemi, V. Presutti, *Ontology Design Patterns*, Handbook on Ontologies, Second Edition, Editors: S. Staab, R. Studer, ISBN: 978-3-540-70999-2, pp.221-244, 2009.
- [4] P. Cimiano, *Ontology Learning and Population from Text. Algorithms, Evaluation and Applications*, Springer, ISBN 978-0-387-30632-2, 2006.
- [5] OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004, <https://www.w3.org/TR/owl-guide/> [retrieved: January 2017].
- [6] A. Piotrowicz, I. Procek, *GUI to Support the Creation of Well-founded c.DnSPL Ontologies* (in Polish), BEng thesis, Poznań University of Technology, Poznań, 2016.
- [7] H. Knoblauch, *An example ontology for tutorial purposes*, mirrored at: <http://users.man.poznan.pl/jolac/travel.owl>, 2016, [retrieved: January 2017]
- [8] Resulting Ontologies, <http://users.man.poznan.pl/jolac/OWLED-ontologies.zip>, 2016 [retrieved: January 2017].
- [9] A. Fernández López, Overview of methodologies for building ontologies. In: Proc. of the IJCAI 1999 Workshop on Ontologies and Problem Solving Methods KRR5 Stockholm, Sweden 1999.
- [10] O. Corcho, M. Fernández López, A. Gómez-Pérez, *Methodologies, tools and languages for building ontologies: where is their meeting point?*, Data & Knowledge Engineering, vol. 46, 1, 41-64, July 2003.
- [11] A. Piotrowicz, I. Procek, *c.DnSPL ontology generation tool*, mirrored at <http://users.man.poznan.pl/jolac/cDnSPL10.jar>, 2016 [retrieved: January 2017].
- [12] J. Cybulka, J. Dutkiewicz, M. Żętkowski, *Ontology-based Generation of Event Extraction Templates and Frames*. In: *Human Language Technologies as a Challenge for Computer Science and Linguistics*, Z. Vetulani & J. Mariani (eds.). In: Proc. of the 7th Language and Technology Conference, Poznań, Poland, <http://lta.amu.edu.pl/book/papers/OWN-2.pdf>, 2015 [retrieved: January 2017].
- [13] J. Cybulka, J. Dutkiewicz, *Events Extractor for Polish in a Semantics-Driven Mode*. In: *Human Language Technologies as a Challenge for Computer Science and Linguistics*, Z. Vetulani & J. Mariani (eds.). In: Proc. of the 7th Language and Technology Conference, Poznań, Poland, <http://lta.amu.edu.pl/book/papers/IRIE-4.pdf>, 2015 [retrieved: January 2017].
- [14] A. Gangemi et al., *C-ODO: an OWL meta-model for collaborative ontology design*, Proc. of Workshop on Social and Collaborative Construction of Structured Knowledge, at 16th International World Wide Web Conference, May 8–12, 2007, Banff, Canada, http://ceur-ws.org/Vol-273/paper_38.pdf, 2007 [retrieved: January 2017].
- [15] B. Stadlhofer, P. Salhofer, A. Durlacher, *An Overview of Ontology Engineering Methodologies in the Context of Public Administration*, Proc. of 7th International Conference on Advances in Semantic Processing (SEMAYRO 2013), September 29-October 3, 2013, Porto, Portugal, ISBN: 978-1-61208-293-6, 2013.
- [16] M. Casellas, *Methodologies, Tools and Languages for Ontology Design*, in: *Legal Ontology Engineering. Methodologies, Modelling Trends, and the Ontology of Professional Judicial Knowledge*, Springer, ISBN 978-94-007-1496-0, pp. 57–109, 2011.
- [17] M.C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández López, *The NeOn Methodology for Ontology Engineering*, in: *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi (Eds.), Springer Berlin Heidelberg, pp. 9–34, 2012.
- [18] RDF Schema 1.1. W3C Recommendation 25 February 2014, <https://www.w3.org/TR/rdf-schema/> [retrieved: January 2017].



Jolanta Cybulka is Assistant Professor at the Institute of Control and Information Engineering of the Faculty of Electrical Engineering at Poznań University of Technology. She received her PhD in Computer Science from the University of Science and Technology (AGH) in Kraków, Poland. Her research, in general, concerns methods and tools of knowledge representation and processing. She has particularly worked on semantics-directed generation of interpreters of programming languages, based on action semantics. Recently she has proposed to model the semantics used in information systems via ontologies which are well-founded, meaning that they represent philosophically and linguistically grounded view on reality.