ESO: a Frame based Ontology for Events and Implied Situations

Roxane Segers
The Network Institute
VU University Amsterdam
r.h.segers@vu.nl

Piek Vossen
The Network Institute
VU University Amsterdam
piek.vossen@vu.nl

Marco Rospocher
Fondazione Bruno Kessler
rospocher@fbk.eu

Luciano Serafini
Fondazione Bruno Kessler
Serafini@fbk.eu

Egoitz Laparra
IXA Group
Universidad del Pais Vasco
Egoitz.Laparra@ehu.es

German Rigau
IXA Group
Universidad del Pais Vasco
German.Rigau@ehu.es

Abstract

This paper presents the Event and Situation Ontology (ESO), a resource which formalizes the pre and post conditions of events and the roles of the entities affected by an event. The ontology reuses and maps across existing resources such as Wordnet, SUMO and Framenet and is designed for extracting information from text that otherwise would have been implicit. We present the metamodel of the ontology and the procedure for building the first version of ESO.

1 Introduction

Technology for detecting events and their participants in text has become widely available, e.g. (Björkelund et al., 2009) and (Das et al., 2010). This technology relies on large-scale language resources with event semantics such as PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998). Within the NewsReader project,[1] we exploit this Semantic Role Labeling (SRL) technology for the purpose of processing large streams of news in different languages. More specifically, we processed millions of English news articles on the global automotive industry covering a period of 10 years, from 2003 till 2013. The processing of such data sets generates hundreds of millions of events involving hundred-thousands of participants, mostly people and organizations. The data enables us to discover long-term developments in this global industry, for example by chaining events over time around particular participants.

Knowledge about what took place in an industry during the financial crisis is useful by itself but it does not show clearly the implications of these events. For example, if there is a management change in a company, we may know that something happened and who is involved but we still do not know the implications of the event. Furthermore, events can be expressed in many different ways (*firing*, *quitting*, *leaving*, *resigning*) and still share the same implications. However, if the implications of an event could be made explicit, we would be able to derive changes in property values for vast amounts of events over long periods of time. Consider for example, the expression 'Y fires X' which implies that X must have been working for Y *before* the firing and that X is not working for Y *after* the firing. Likewise, the expression 'X works for Y', states that some situation holds *during* some period of time. Deriving these implications demands a model that a) defines what the implications of events are; b) defines what entities are affected by an event and c) defines how the implications of dynamic and static events can be linked. Though work on deductive reasoning over Frame annotated text (e.g. (Scheffczyk et al., 2006)) and defining pre and post situations of predicates exist ((Im and Pustejovsky, 2009) and (Im and Pustejovsky, 2010)), to the best of our knowledge, ontologies that model all three aspects do not exist. Most closest comes the extension to DOLCE-LITE (Hicks, 2009) that was developed in the KYOTO project.[2] In this ontology, property values are modeled as quality regions for reasoning. However, these quality regions are not connected to the events in the ontology as pre and post conditions. Finally, the axioms and definitions provided in generic ontologies aim at providing a comprehensive semantic specification of the concepts. There is no guarantee that these specifications also capture the kind of changes that are relevant for our domain and they most likely capture many implications that are not relevant. Reasoning using generic ontologies can thus be seen as undirected reasoning without a guaranteed out-

[1] www.newsreader-project.eu

[2] http://kyoto-project.eu

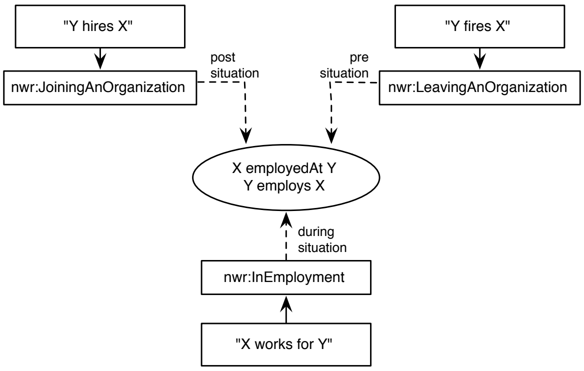

Figure 1: Dynamic and static event expressions and their shared situation.

come.

We therefore developed the Event and Situation Ontology to enhance the extraction and linking of dynamic and static events and their implications in text.[3] By explicitly modeling the implications pertaining to events, we can extract sequences of states and changes over time regardless of if this information was directly expressed in text, or inferred by some reasoner. Figure 1 shows part of our model for expressions for dynamic (*hire*, *fire*) and static events (*works for*) and their associated situation (at the centre of the figure).

The development of ESO was carried out in the background of the Newsreader project. The Predicate Matrix (de Lacalle et al., 2014), that integrates predicate and role information from FrameNet, VerbNet, PropBank and Wordnet, is used to assign role and predicate annotations on document level. All definitions and assertions in ESO are fed back to the Predicate Matrix. As such the ontology provides an additional layer of annotations that allows inferencing over events and implications.

Following best practices in Semantic Web technologies, ESO reuses parts of two existing vocabularies: there are mappings from our ontology to Framenet on class and role level and mappings to SUMO on class level. As such, we can define our classes without adhering to modeling choices in Framenet and SUMO. Through these mappings, ESO serves as a hub to other vocabularies as well, such as Princeton Wordnet (PWN) and the Wordnets in the Global Wordnet grid. As such, ESO provides a domain-specific hierarchy of events and their implications across languages.

The remainder of this paper is organized as follows; section 2 presents the ontological meta model; section 3 elaborates on building ESO and the different modeling choices; in section 4 we discuss the integration of ESO and the Predicate Matrix; in section 5 we show how ESO is applied to our document collection. We conclude in section 6 with a discussion and some outlines for future work.

2 Ontological meta model

The ontology for NewsReader should enable us to retrieve property values over time for entities in large data sets. We should be able to query the data set for all the companies a person worked for in the last 10 years and when, or what managers worked for a company in the last 10 years and in which periods. We cannot rely on the fact that this information is directly reported in the news articles that we process. In many cases, it is only indirectly expressed by the events and changes mentioned. Given this, we can formulate the following requirements for the ontology:

- model properties and values
- model the pre and post conditions that apply to the events
- model roles of entities for the events in relation to these pre and post conditions
- model the reasoning to derive situations from pre and post conditions associated with the events and roles

We next describe the ontology meta model to meet these requirements.[4] In addition, we need to map the words that we find in the news to event and property classes. We also need to map the semantic roles assigned by the SRL module to the roles in ESO. Finally, we want to be able to transfer these mappings from English to other languages. These mappings are discussed in sections 3 and 4.

ESO is an OWL 2 ontology.[5] It assumes that the semantic representation of text is converted to an RDF representation of event and entity instances between which relations are expressed as triples. Such a representation is schematically represented as follows:

[3]The ontology can be accessed at http://www.newsreader-project.eu/results/event-and-situation-ontology/

[4]A more comprehensive description of the model and its instantiating can be found online: `http://www.newsreader-project.eu/`

[5]`http://www.w3.org/2001/sw/wiki/OWL`

```
:obj-graph-eventX {
    :eventX
        a     nwr:LeavingAnOrganization;
        nwr:LeavingAnOrganization_employee    :employeeA;
        nwr:LeavingAnOrganization_employee    :employeeB;
        nwr:LeavingAnOrganization_employer    :companyA;
        sem:hasTime  :time_eventX.
}
```

These statements specify that the event is of a certain type (nwr:LeavingAnOrganization), that it involves a entity playing the role of an employer (:companyA) and two entities playing the role of employees (:employeeA,:employeeB), and that it occurred at a certain time (:time_eventX). From these representations, we need to derive the statements that express the pre event states and post event states applying to the entities involved. For this purpose, we defined five core classes in ESO, which are further specialized in subclasses:

Event: this class is the root of the taxonomy of event types. Any event detected in a text will be an instance of some class of this taxonomy;

DynamicEvent: this is a subclass of Event (for which dynamic changes are defined) that apply to FrameNet frames that can be considered as dynamic events (e.g. fn:Firing);

StaticEvent: this is another subclass of Event for "static" event types which capture more static circumstances (e.g., fn:Being_employed, fn:Being_located); they typically directly trigger a situation holding at the time the event occurs (a "during situation").

Situation: the individuals of this class are actual pre/post/during situations that will be instantiated starting from the event instances detected in the text;

SituationRule: the individuals of this class enable to encode the rules for instantiating pre/post/during situations when a certain type of event is detected;

Analogously to FrameNet frame elements for frames, ESO represents the role of an entity in an event. Roles are formalized as object properties: this way, an event instance :eventX can be related to an entity :entityZ participating in it with assertions of the form:

:eventX nwr:hasRoleY :entityZ

where nwr:hasRoleY specifies the role of :entityZ in :eventX. Each object property defining a role in the ontology is defined as subproperty of the top object property nwr:hasRole: this way, given any event, we can retrieve the entities participating in it by looking at assertions having as predicate the property nwr:hasRole.

2.1 Formalization of the rules for instantiating situations from events

Each nwr:SituationRule individual is specialized to define exactly how the triples inside the Situation named graph have to be defined. This is done by defining an individual for each assertion to be created, which has three annotation properties assertions:

nwr:hasSituationAssertionSubject: the object of this triple is the role of the event to be used as subject in the assertion, e.g. 'employment-employee';

nwr:hasSituationAssertionProperty: the object of this triple is the predicate to be used in the assertion. It is either a binary property or an unary property, e.g. 'employedAt';

nwr:hasSituationAssertionObject: the object of this triple is the role of the event to be used as object in the assertion, e.g. 'employment-employer'.

Specific nwr:SituationRules are defined that are triggered as the pre or post situation value related to situations of particular classes of events. By triggering these rules, a reasoner can thus infer from the fact that a particular situation belongs to the class LeavingAnOrganization and has entity instances in certain roles that a) one of these entities at some point in time was employed by the other and b) after that point in time it was not. Consider for instance the nwr:pre_ChangeOfPossession situation rule:

```
nwr:pre_ChangeOfPossession
    nwr:hasSituationRuleAssertion    pre_ChangeOfPossessionA1;
    nwr:hasSituationRuleAssertion    pre_ChangeOfPossessionA2.
```

This rule triggers the instantiation of two assertions, nwr:pre_ChangeOfPossessionA1 and nwr:pre_ChangeOfPossessionA2, defined as follows:

```
nwr:pre_ChangeOfPossessionA1
    nwr:hasSituationAssertionSubject     nwr:possessionOwner1;
    nwr:hasSituationAssertionProperty    nwr:possess;
    hasSituationAssertionObject          nwr:possessionTheme.

nwr:pre_ChangeOfPossessionA2
    nwr:hasSituationAssertionSubject     nwr:possessionOwner2;
    nwr:hasSituationAssertionProperty    nwr:notPossess;
    hasSituationAssertionObject          nwr:possessionTheme.
```

Therefore, from an event instance :eventX of type nwr:ChangeOfPossession, having roles :instanceX (nwr:possessionOwner1 role), :instanceY (nwr:possessionOwner2 role), and :instanceZ (nwr:possessionTheme role), by interpreting the aforementioned rule schema we can instantiate a pre-situation named graph, :eventX_pre, defined as follows:

```
:eventX_pre {
    :instanceX    nwr:possess       :instanceZ .
    :instanceY    nwr:notPossess    :instanceZ .
}
```

where the first assertion is created due to nwr:pre_ChangeOfPossessionA1, and the second assertion to nwr:pre_ChangeOfPossessionA2. In addition to rules that connect static relations to dynamic changes, the ontology also models the temporal implications. The time at which the dynamic situation takes places thus marks the beginning or ending of the static situation. For further details on the derived temporal implications, we need to refer the reader to (van Erp et al., 2014) due to space limits.

2.2 Mappings from external resources to ESO

A key ingredient of the ESO ontology is the mapping of the FrameNet frames and frame elements to the event types and roles that we defined. This mapping is necessary to translate the role annotations provided by the SRL module to our ontology vocabulary, exploited by the reasoning module to instantiate situations from events.

For each event type (modelled as class in the ontology) and each role (modelled as object property) we defined some annotations (nwr:correspondsToFrameNetFrame and nwr:correspondsToFrameNetElement), representing the corresponding frames and frame elements. For instance, nwr:Giving has three annotations via nwr:correspondsToFrameNetFrame to frames fn:Giving, fn:Sending, and fn:Supply, meaning that if a frame of type fn:Supply or any of the others is identified in the text, it has to be considered as an event of type nwr:Giving, and therefore pre and post situation rules defined for nwr:Giving should be triggered. Similarly, the role nwr:possession-owner_1 is mapped via nwr:correspondsToFrameNetElement assertions to frame elements: fn:Seller, fn:Supplier, fn:Lender, fn:Sender, fn:Donor, fn:Source, fn:Agent, fn:Exporter, and fn:Victim. We also defined mappings from the ESO event types to SUMO[6] classes via nwr:correspondsToSUMOClass annotation assertions.

These mappings make clear that our ontology is providing only a partial definition for concepts. We only need to capture the implications for the reasoning as a result of the pre and post condition states related to the events and not all other aspects of meaning. As such there is no difference for the ownership implication if somebody takes something or steals it.

3 Building ESO

In this section, we describe in more detail how the different components of the ontology were build. As input data for ESO, we used a collection of 63K English text documents that were processed by our NLP pipeline in a first run. To obtain the most salient event predicates from this collection, we annotated all predicates with a FrameNet mapping as being contextual (*increase, hire*) or related to communication (*say*), cognition (*think*) or perception (*see*). Only the contextual predicates were used for building the ontology. Thus, we started with 234 frames and 1306 unique predicates potentiality important for the domain. To scope this set, we discarded low frequent frames and those frames that were clearly nor related to the car domain e.g. fn:Cooking_creation. As a result, an unstructured set of 92 frames remained.

First, we tried to conceptually select and group the car frames for the ontology by using the FrameNet inheritance relations. This turned out to be problematic as FrameNet has no full subclass hierarchy. Also, some frames are associated with lexical units that represent different concepts from a more ontological point of view. For instance fn:Forming_Relationships groups both *marry* and *divorce* into a frame while different pre and post situations will hold. For deriving a conceptual structure for the frames, we therefore decided to use the SUMO ontology[7] as a background model as it is freely available, well-documented, it has a good coverage, and it is mapped to the English Wordnet. The event hierarchy for ESO was derived in five steps.

Step 1. The unstructured set of 92 frames was mapped manually to SUMO classes by means of subclass and equivalence relations. We define a frame and all its content as a class denoting a concept of change or state for which at least one basic implication should hold. If due to different modeling choices between SUMO and FrameNet a near equivalence, subclass or superclass mapping was not possible, we stored the frame as candidate class without mapping.

Step 2. From the mappings, we selected four top nodes in SUMO that represented the main conceptual clusters for the frames express-

[6] `http://www.ontologyportal.org`

[7] `www.ontologyportal.org`

ing events: Motion, InternalChange, ChangeOfPossession and IntentionalProcess. In this step, we also started to group similar frames into one class. For instance, the main difference between the frames Departing and Quitting_a_place is a specification of the entity that moves. For our purposes, this level of granularity is not necessary. As such, both frames have been defined as corresponding to the ESO class Departing and SUMO class Leaving.

Step 3. Next, we checked the SUMO class hierarchy of Motion, InternalChange, ChangeOfPossession and IntentionalProcess to select additional classes that may be of importance for the car domain, such as Investing and Importing.

Step 4. Based on this, we built four initial sub hierarchies consisting of ESO classes with a mapping to SUMO and FrameNet, and potential ESO classes with only a mapping to SUMO.

Step 5. To increase the coverage, we manually mapped back from ESO classes to FrameNet frames. For this, we used the existing frame-to-frame relations in FrameNet (Ruppenhofer et al., 2006). These additional frames were either a) found in the document collection, but previously ruled out by the thresholds or, b) not found in the car data but a frame for the ESO class does exist in FrameNet. In some cases, frames were found for which we had no SUMO-based ESO class. In those cases, a new ESO subclass was defined. Also, for some SUMO-based ESO classes no corresponding frame could be found. These classes were kept in the ontology nonevertheless as placeholder for future extensions. As such, we have ESO classes with mappings to both FrameNet and SUMO, ESO classes with only a mapping to FrameNet, and ESO classes with only a mapping to SUMO. Furthermore, to keep the hierarchy clean, we opted to use single inheritance only for all event classes in the ontology.

3.1 Properties for defining pre and post situations of an event

The second component of the ontology consists of properties for pre and post situations that state which situation holds before and after an event. All situation properties were hand build based on the shared semantics of the predicates related to a frame. In this version of the ontology only one salient property is defined, but the model allows for additional properties. We opted to define situation properties at the highest class as possible to allow the inheritance of properties by all subclasses.

Since we generalize over different lexicalizations and syntactical structures of events, it is in some cases difficult to define properties for pre and post situations at the conceptual level. Without context, some classes are interpretable as both dynamic and static events which has an effect on what situations hold before or after the event. The class 'Borrowing' for instance, can be the *moment* of a change of possession (dynamic event) or the *time span* someone has an entity in possession (static event). For this version of the ontology, we defined no pre and post situations for this type of class.

Another issue is that some properties currently may be either too strong or to weak. By intuition, a property such as 'hasInPossession' should hold for the class ChangeOfPossession and all its subclasses. However, the semantics of this property seems weakened from actual 'ownership' into a more vague 'having' for a class such as nwr:Taking. From the results of the reasoner, we can finetune these implications if necessary.

For static events, we defined situation properties that are true for the duration of the static event. For the class nwr:InEmployment the property employedAt defines that some person is employed at some employer. The same property is used as pre situation for the event class nwr:LeavingAnOrganization and the post situation for the class nwr:JoiningAnOrganization. Where applicable, related events and situations have been related as such by means of a shared situation property. As a result, the relation between an inferred situation of a dynamic event and the more explicit situation of a static event becomes more easily understandable.

3.2 Roles for the entities affected by an event

For the roles of the entities that are affected by an event, we used a selection of FrameNet Frame Elements (FEs), derived from the mappings of an ESO class to FrameNet frames. Thus, we define which roles are important for modeling our domain and which roles are not. For the class nwr:Translocation and all its subclasses, the entity that translocates maps to FrameNet FE Self-mover, Theme and Driver and the entity that expresses the location to the FEs Location and Goal.

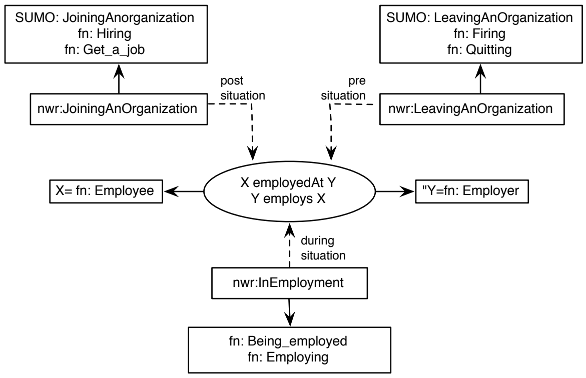

Figure 2: Dynamic and static events, situations and mappings to SUMO and FrameNet

Other less salient FEs that are specified in translocation frames such as Manner, Distance and Speed are not incorporated. Explicit and necessary role selection also scopes the pre and post situations to events: *running around naked* (no source or goal) is ruled out as dynamic event, while the formulated pre and post situations will hold for *he is running to Rome* (goal).

3.3 Event and Situation Ontology version 1

The first version of the ESO now consists of 59 event classes divided over dynamic events (50) and static events (9). The dynamic event class hierarchy consists of four major nodes: ChangeOfPossession (16 subclasses), Motion (10 subclasses), InternalChange (11 subclasses) and IntentionalEvents (11 subclasses). For 53 classes we have one or multiple mappings to FrameNet frames. In total, 94 mappings to FrameNet were made, covering 532 unique combinations of a predicate and a frame. Additionally, 49 out of 59 event classes have a mapping to SUMO. Furthermore, we defined 24 properties (20 binary and 4 unary) such as 'atPlace', 'employedAt' and 'hasInPossession' which define the situations statements for 35 out of 50 dynamic event classes and all 9 static event classes. Finally, we defined 33 different roles for the entities affected by an event or situation. Each role is mapped to one or more Frame Entities in FrameNet (60 mappings in total).

The interplay between static and dynamic event classes, situations and mappings to SUMO and FrameNet is presented in figure 2.

4 Integrating ESO into the Predicate Matrix

The previous sections explained how to integrate FrameNet and the Domain Ontology to obtain ESO representations of the SRL output. Obviously, this translation is only available for annotations based on FrameNet while the SRL modules use PropBank predicate models. For this purpose, we can make use of the interoperable capabilities offered by the PredicateMatrix (PM) (de Lacalle et al., 2014). The PM is an automatic extension of SemLink (Palmer, 2009) that merges several models of predicates such as VerbNet (Kipper et al., 2000), FrameNet (Baker et al., 1998), PropBank (Palmer et al., 2005) and WordNet (Fellbaum, 1998). The PM also contains for each predicate features of the ontologies integrated in the Multilingual Central Repository (Gonzalez-Agirre et al., 2012) like SUMO (Niles and Pease, 2001), Top Ontology (Álvez et al., 2008) or WordNet domains (Bentivogli et al., 2004). The mappings between such knowledge bases allow to take advantage from their individual strengths. For example, the coverage of PropBank or the semantic relations among events and participants of FrameNet. Moreover, it is also possible to automatically integrate all knowledge connected to any of its components, as in the case of ESO.

Both FrameNet and SUMO labels integrated in ESO are used to connect ESO to the PM. For example, the predicate **sell.01** of PropBank belongs, according to their mappings in the PM, to the frame *Commerce_sell* of FrameNet. Thus, this predicate and its arguments would be mapped to ESO as shows table 1. Moreover, the frame can also be linked through the SUMO classes. For instance, the predicate **drain.01** of PropBank belongs to the frame *Emptying* that is not considered in ESO. However, it also belongs to the class *Removing* of SUMO and, in consequence, the mappings in table 2 can be obtained. In this way, ESO is connected to **2235** predicates and **3445** different roles of the PM.

PB-pred	PB-arg	FN-frame	FN-fe	ESO-class	ESO-role
sell.01	arg_0	Commerce_sell	Seller	Selling	possession-owner_1
sell.01	arg_1	Commerce_sell	Goods	Selling	possession-theme
sell.01	arg_2	Commerce_sell	Buyer	Selling	possession-owner_2

Table 1: Mapping between PropBank and ESO through FN.

PB-pred	PB-arg	SUMO-class	FN-fe	ESO-class	ESO-role
drain.01	arg_0	Removing	Theme	Removing	translocation-theme
drain.01	arg_1	Removing	Source	Removing	translocation-source

Table 2: Mapping between PropBank and ESO through SUMO.

5 Current Output of the NWR system

PredicateMatrix version 1.1 integrating the ESO information is now used by the SRL module of NewsReader. This means that any possible mapping of a word to an ESO class, either through WordNet, FrameNet or SUMO is added as an external reference to a predicate. The same applies to the roles that are associated with the predicate. For the following sentence ”Council President Bob Reiss directed Boran to hire Lizzano and fire Shook”,[8] NewsReader will generate SRL structures for the predicate *hire* and involving *Boran* and *Lizzano* as shown in table 3. The SRL structure shows both the corresponding FrameNet frames and elements as the ESO classes and roles to which they are mapped.

Expression	FrameNet	ESO
hire	fn:Hiring	nwr:JoiningAnOrganization
Boran	fn:Hiring@Employer	nwr:JoiningAnOrganization@employment-employer
Lizzano	fn:Hiring@Employee	nwr:JoiningAnOrganization@employment-employee

Table 3: SRL output using the PredicateMatrix mappings

From the SRL structures and various other annotations generated in NewsReader, we create an RDF representation of the situation as a named graph including all the stated triples with unique URIs for instances of events and entities:

```
nwr:namedGraph47P9-DCM0-0092-K267.xml
#pr41rl98 {
    nwr:47P9-DCM0-0092-K267.xml#hireEvent
     a     nwr:JoiningAnOrganization;
nwr:JoiningAnOrganization@employment-employer
       nwr:47P9-DCM0-0092-K267.xml#Walter_Boran ;
nwr:JoiningAnOrganization@employment-employee
       nwr:47P9-DCM0-0092-K267.xml#Lizzano> ;
sem:hasTime
       nwrtime:200211.}
```

We thus state that the event is an instance of the class nwr:JoiningAndOrganization, what the ESO role relations are with the entities and what the time was of the event. The time URI is an instance represented in normalized form according to owl-time. In this example, it points to the month November 2002. Given this named graph, ESO can be used to derive static situations by triggering the rules. This will generate the following graph:

[8] 2003/01/15/47P9-DCM0-0092-K267.xml

```
nwr:namedGraph47P9-DCM0-0092-K267.xml
#pr41rl98assertion1 {
    nwr:47P9-DCM0-0092-K267.xml#employment
     a     nwr:inEmployment;
nwr:JoiningAnOrganization@employment-employer
   nwr:47P9-DCM0-0092-K267.xml#Walter_Boran ;
       nwr:JoiningAnOrganization@employment-employee
   nwr:47P9-DCM0-0092-K267.xml#Lizzano>  ;
time:hasBeg
       nwrtime:200211.}
```

Here we state that there is an instance of a static situation of the type nwr:inEmployment that involves Walter Boran as employer and Lizzano as employee and that it begun at time nwr:200211.

Since collections of triples are grouped as named graphs, we can express properties of these collections through other triples. This is used within the project to indicate the provenance of the triples. In this case, we state that the former graph with the URI nwr:namedGraph47P9-DCM0-0092-K267.xml#pr41rl98 is attributed to the author and publisher of the news article, whereas the derived graph with the URI nwr:namedGraph47P9-DCM0-0092-K267.xml#pr41rl98assertion1 is attributed to the ontology and a version of the reasoner that has been applied.

```
nwr:namedGraph47P9-DCM0-0092-K267.xml#pr41rl98
         prov:attributedTo
                nwr:47P9-DCM0-0092-K267.xml#William_J_Ford ,
                http://www.mcall.com;

nwr:namedGraph47P9-DCM0-0092-K267.xml#
pr41,rl98assertion1
         prov:attributedTo
                nwr:ontology/eso ,nwr:reasoner/version1;
```

We are processing over a million of news articles on the automotive industry. We took the news published in 2003 for estimating the coverage of ESO with respect to the predicates and roles detected.

Nr. Files	137,947	Freq/file
predicates	9,186,927	66.6
roles	20,095,946	145.7
fn frames	8,943,414	64.8
fn elements	16,612,989	120.4
ESO types	2,112,609	15.3
ESO roles	2,163,377	15.7

Table 4: Ratios of predicates, roles, frames, frame elements and eso mappings, for 137,947 articles published in 2003

On average, there are about 66 predicates and 145 roles per news article. If we compare this with the average FrameNet frames and elements, we see that the proportion are 97.35% for frames versus predicates and 82.67% for elements versus roles. If we consider the ESO types and roles, we see that these proportions are 23.00% and 10.77% respectively. Obviously, ESO was developed for a small set of concepts and roles only so we cannot expect full coverage.

6 Conclusion and Future Work

In this paper, we presented the Event and Situation Ontology for inferencing on event data automatically extracted from large streams of news. The ontology was designed to capture implications of events with respect to a selected set of properties only. It thus does not provide a complete definition of events and situations.

We are currently processing over a million of news articles on the automotive industry, where the ESO mapping are inserted in the SRL layers. The output will be converted to RDF, after which we will apply reasoning to derive new statements. The output will be evaluated through inspecting samples, against benchmark data that is being developed and through end-user tasks on the data sets.

Furthermore, we are planning to extend and improve the mappings across FrameNet, WordNet and ESO as well as extending ESO to other properties, such as numerical and scalar values.

Acknowledgments

The research for this paper was supported by the European Union's 7th Framework Programme via the NewsReader Project (ICT-316404).

References

J. Álvez, J. Atserias, J. Carrera, S. Climent, A. Oliver, and G. Rigau. 2008. Consistent annotation of eurowordnet with the top concept ontology. In *Proceedings of GWC'08*.

C.F Baker, C.J. Fillmore, and J.B. Lowe. 1998. The berkeley framenet project. In *Proceedings COLING-ACL*, ACL '98, Montreal, Canada.

L. Bentivogli, P.Forner, B. Magnini, and E. Pianta. 2004. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Proceedings of the Workshop on Multilingual Linguistic Ressources*. Association for Computational Linguistics.

A. Björkelund, L. Hafdell, and P. Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of CoNLL-2009*, Boulder, CO, USA.

D. Das, N. Schneider, D. Chen, and N.A. Smith. 2010. Probabilistic frame-semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10.

M. Lopez de Lacalle, E. Laparra, and G. Rigau. 2014. First steps towards a predicate matrix. In *Proceedings of GWC 2014*, Tartu, Estonia.

Christiane Fellbaum. 1998. *WordNet: an electronic lexical database*. MIT Press.

A. Gonzalez-Agirre, E. Laparra, and G. Rigau. 2012. Multilingual central repository version 3.0. In *LREC*, pages 2525–2529.

A. Hicks. 2009. Domain extension of the central ontology - final. Deliverable 8.3, KYOTO-ICT 211423.

S. Im and J. Pustejovsky. 2009. Annotating event implicatures for textual inference tasks. In *Proceedings of the 5th International Conference on Generative Approaches to the Lexicon*.

S. Im and J. Pustejovsky. 2010. Annotating lexically entailed subevents for textual inference tasks. In *Proceedings of FLAIRS-23*, Daytona Beach, USA.

K. Kipper, H. Trang Dang, and M. Palmer. 2000. Class-based construction of a verb lexicon. In *Seventeenth National Conference on Artificial Intelligence*, AAAI-2000.

M.Palmer, D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

I. Niles and A. Pease. 2001. Towards a standard upper ontology. In *Proceedings of FOIS-Volume 2001*. ACM.

Martha Palmer. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the Generative Lexicon Conference*, pages 9–15.

J. Ruppenhofer, M. Ellsworth, M. Petruck, C.R. Johnson, and J. Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*. International Computer Science Institute, Berkeley, California.

J. Scheffczyk, A. Pease, and M. Ellsworth. 2006. Linking framenet to the suggested upper merged ontology. In *Proceedings of FOIS 2006*.

M. van Erp, R. Segers, P. Vossen, A. Fokkens, M. Rospocher, and R. Agerri. 2014. D5.2.1 domain model for financial and economic events, version 1. Deliverable D5.2.1, NewsReader-ICT316404.