

Toward the Automation of Business Process Ontology Generation

Sergio de Cesare, Damir Juric and Mark Lycett
Brunel University
Uxbridge, UB8 3PH, United Kingdom
{sergio.decesare|damir.juric|mark.lycett}@brunel.ac.uk

Abstract—Semantic Business Process Management (SBPM) utilises semantic technologies (e.g., ontology) to model and query process representations. There are times in which such models must be reconstructed from existing textual documentation. In this scenario the automated generation of ontological models would be preferable, however current methods and technology are still not capable of automatically generating accurate semantic process models from textual descriptions. This research attempts to automate the process as much as possible by proposing a method that drives the transformation through the joint use of a foundational ontology and lexico-semantic analysis. The method is presented, demonstrated and evaluated. The original dataset represents 150 business activities related to the procurement processes of a case study company. As the evaluation shows, the proposed method can accurately map the linguistic patterns of the process descriptions to semantic patterns of the foundational ontology to a high level of accuracy; however further research is required in order to reduce the level of human intervention, expand the method so as to recognise further patterns of the foundational ontology and develop a tool to assist the business process modeller in the semi-automated generation of process models.

Keywords— *business process modelling; foundational ontology; lexico-semantic analysis; ontology construction; perdurantism; BORO*

I. INTRODUCTION

Business Process Management (BPM) is a discipline that includes modelling, management, execution and analysis of business processes. With many tasks in the BPM lifecycle still manual and cumbersome to execute, Semantic Business Process Management (SBPM), or the adoption of semantic technologies in BPM, can help to improve and automate the lifecycle via semantically richer data representations and querying capabilities. While SBPMN offers promising prospects, organisations wishing to adopt such technologies would require an initial effort in mapping existing process representations to ontological models and maintain these models and the implemented data repository over time. In modern organisations this means managing an ever-increasing amount of data deriving, in most cases, from different sources including textual and graphical specifications. Moreover ontological expertise is required to carry out such re-engineering.

This paper presents a method for automatically generating ontological representations from the textual descriptions of business process maps; thus providing a way to potentially reduce the time and cost of carrying out the initial transformation required to produce business process ontologies from existing models and data representations.

Hepp and Roman [1] distinguish between three types of ontologies which they consider necessary for SBPM. These are process, organisational and domain-specific ontologies. Process ontologies primarily describe the structure of a process, while organisational ontologies describe the artefacts that are involved in process realisation. Domain ontologies contain information specific to each organisation. In this paper we focus on the automated construction of process ontologies.

A distinctive feature of the method proposed in this paper is its grounding in a foundational ontology used to drive the lexico-semantic analysis of the source models. The foundational ontology adopted is the Business Object Reference Ontology (BORO) [2]. This four-dimensional worldview allows for more accurate semantic representations of processes and their elements such as temporal whole-part relationships, typical when modelling processes. An example of four-dimensional modelling in an enterprise environment can be seen in [4].

In recent years, with a huge amount of available digital content, methodologies have been developed for automatically creating and populating ontologies from text sources. However, to the best of our knowledge, this is the first attempt at automatically discovering semantic models from textual descriptions by driving the lexico-semantic analysis with a foundational ontology.

This paper presents an unsupervised method for constructing business process ontologies automatically from the available organisational data. The paper is structured as follows. Section II presents related work. Section III describes our dataset and Section IV and V present the algorithm for automating the creation of 4D business process ontologies. The final sections evaluate the algorithm and present findings and problems specific to our automated method.

II. RELATED WORK

Ontology construction is normally carried out manually but in recent years automated approaches have emerged. Most of these approaches deal with raw text, but some also use other sources such as Wikipedia pages and HTML forms. This section summarises a few of these different approaches. More closely related to our work are those approaches that process semi-structured text (e.g., workflow descriptions).

Manually developed general ontologies are still the most widely used type of ontologies [5, 6, 7]. The construction of such ontologies is a very expensive and time-consuming process. Moreover, the process of acquiring new knowledge is always needed and it requires ongoing work by human experts, even after the ontology has been released. In order to solve the problem of human workforce, some techniques propose a broader collaboration during the ontology construction process, as in the case of Semantic Wikipedia [8], where facts are created and incorporated into an ontology by many volunteers.

As for automated approaches, Zhou [9] gives a typical scenario of an ontology learning process (which can either be manual or automated) and it consists of: creating concepts, creating relations, ontology population and ontology evaluation. Wiszniewski [10] introduces a metamodel for ontology learning from text and presents an extensive survey of ontology learning models. A review of the literature shows that three different approaches for extracting possible binary relations from text sources have been developed, each of them giving its primary focus to different features of the source text: collocations, syntactic dependencies or lexico-syntactic patterns. Many developed techniques are a combination of these approaches [11, 12, 13, 14, 15, 16]. We mention here that our approach uses syntactic dependencies. An interesting project is described in [17]. ConceptNet is a large semantic graph that describes general human knowledge and how it is expressed in natural language by collecting data from different sources.

The approach described by De Nicola et al. [18] is the closest to ours in the sense that the authors deal with the specific domain of business processes. They describe the phases, steps, and intermediate outcomes of the UPON method (Unified Process for ONtology), designed in accordance with the software engineering UP method, aimed at guiding ontology engineers in the production of an effective and valuable domain ontology. The study presents findings on a case study from the e-business domain. While authors came up with a solid methodology for ontology construction in the business domain, all the work is done manually (especially the cumbersome process of storyboard analysis and vocabulary creation). Continuation of this work can be seen in [19] where the ontology modelling framework is presented and aimed at supporting business experts in building an ontology. This work provides a limited number of high-level conceptual templates conceived along the lines of software design patterns. Those templates contain entities like actor, object and process (entities that we extract automatically using syntactic patterns). In [20]

the authors describe the semantic extension of event-driven process chains. They specify the semantics of individual modelling elements by using labels in natural language. Using natural language makes this approach similar to ours but without the automatic generation of vocabulary. Our approach uses textual descriptions of workflows. There has been an increased interest in an analysis of business process patterns specifically in the form of workflows in the last few years due to the emergence of the service-oriented paradigm in which workflows are composed by orchestrating or choreographing Web services. van der Aalst et al. [21], for example, produced a set of workflow patterns. However, as Thom et al. [22] point out, these workflow patterns are relevant toward the implementation of workflow management systems rather than identifying business activities that a modeller can consider repeatedly in different process models. In fact the workflow patterns of van der Aalst et al. [21] are similar to patterns of reusable control structures (for example, sequence, choice and parallelism) rather than patterns of reusable business processes subject to automation (for example, the generic process of developing a new product line). As such these patterns do not resolve the problems of domain reuse in modelling organisational processes [23].

III. BORO FOUNDATIONAL ONTOLOGY

As mentioned earlier, this research adopts a 4D foundational ontology called BORO [2]. The reason for adopting such an ontology is its suitability to modelling processes and the fact that a 4D theory of existence overcomes the traditional dichotomy between dynamic and static modelling, hence the same model can easily represent processes and things that are not traditionally considered as processes (e.g., people, products, machines, etc.)

In BORO identity of individual objects (such as ‘the person John Smith’, ‘John’s car’ or ‘the manufacturing process currently running in our London plant’) is defined by the object’s spatiotemporal extension (or extent), i.e. the space and time that the object occupies in the universe. Therefore the whole of an individual object starts when the object is created (e.g., birth of a person or the start of a process) and ends when the object ceases to exist (e.g., death of a person or termination of a process). Change in the 4D paradigm is defined as the succession of dissimilar temporal parts; for example, a person would traverse phases such as childhood, teenagehood and adulthood; a specific process traverses so-called activities. Each phase or activity in these examples is called a temporal part. In BORO there are two types of temporal parts: states, which have duration (for example, ‘John’s childhood’ or ‘activityX of processY’), and events, which are instantaneous. Individual objects including states are bounded by a creation event (e.g., birth) and a dissolution event (e.g., death).

The relationship between a whole and its temporal parts is called *temporalPartOf*. This relationship type has two subtypes specifically used for events; these are *happensTo* and *happensAt*. The former is specialised in *creates* and *dissolves*

representing the relation between a *CreationEvent* or the *DissolutionEvent* and the created or dissolved individual object respectively. The latter relates an event with the *TimeInstantorPeriod* in which the event takes place.

Figure 1 provides a partial view on the BORO foundational ontology necessary to understand the remainder of the paper. BORO’s top-level categories are individuals, types (or classes) and tuples. As with individuals, extensionalism defines the

identity of classes and tuples in the following manner: the identity of a class is the set of its instances, whereas the identity of a tuple is given by the places occupied by the objects in the relation; for example the class *Cars* is the set of all cars that have existed in the past, that currently exist and that will exist; the tuple *(William, Charles)* represents the relation in which *William* occupies the first place in the tuple and *Charles* the second place. This specific tuple would be an instance of the *sonOf tupleType*.

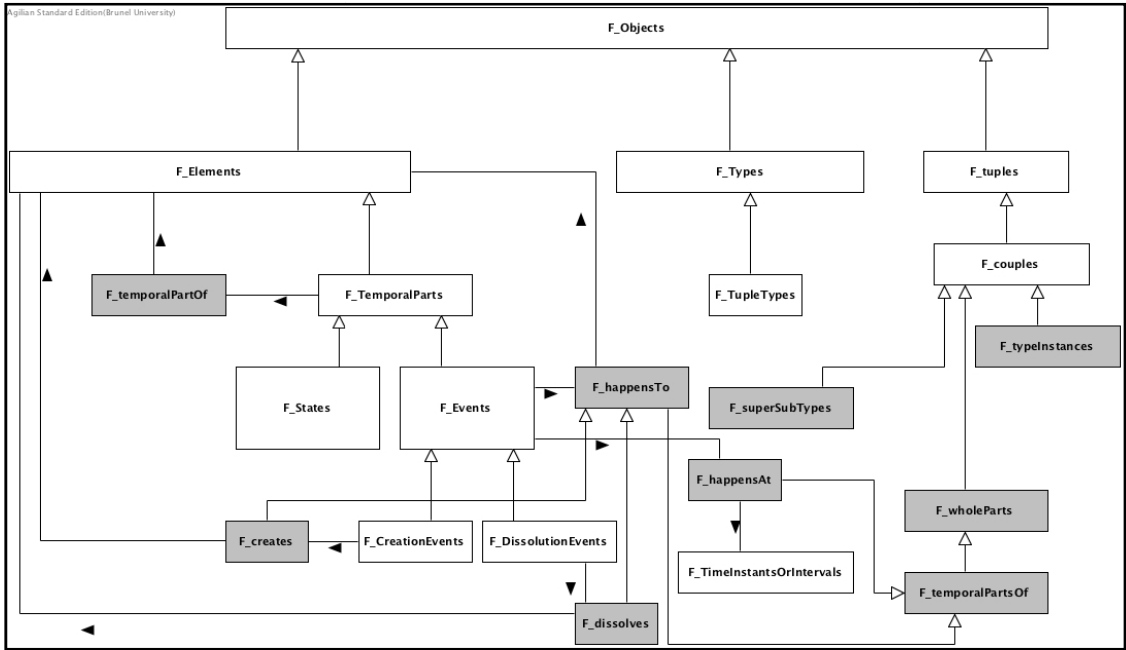


Figure 1. BORO Foundational Ontology (Partial).

IV. DATASET ANALYSIS AND ONTOLOGY DEVELOPMENT

In our research we aim to use so called ‘empirical data’, data produced and used by organisations on an everyday basis and representing their business. Different organisations have different ways for storing or representing their business data.

The dataset of this research came in the form of workflow representations. Graphical representations were accompanied with textual descriptions of business activities together with the roles in charge of each activity. 29 different business processes were identified together with descriptions of 150 business activities.

Figures 2 and 3 represent a very common process of purchasing goods in a large organisation. The process begins with a buyer initiating a requisition and producing a purchase

order. The purchase order becomes an official contract when it is approved by the seller (who sends an invoice). Besides the buyer, other roles in this process can include inventory manager or accounts payable (AP) manager. Most of the process consists of different activities carried out by the previously mentioned roles.

<ul style="list-style-type: none"> Activity 5: Pay suppliers, post payments & settle account
<ul style="list-style-type: none"> When (Input): Matched posted invoices for payment / applied credit notes
<ul style="list-style-type: none"> When (Input): Approved expense claim in line with policy
<ul style="list-style-type: none"> Why (Output): Competitively sourced goods and services
<ul style="list-style-type: none"> Who: AP Mgr

Figure 2. Textual description of a business process.

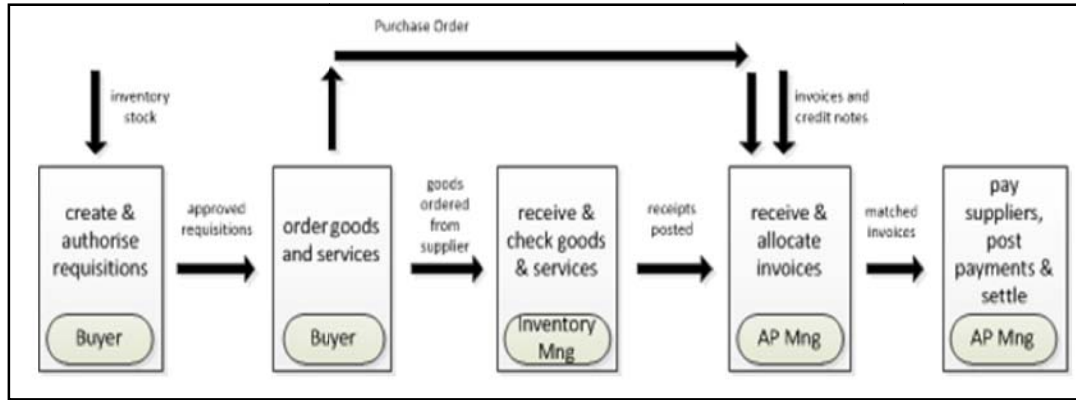


Figure 3. Graphical representation of a purchase order.

The Ontology Development process must first consider how the process descriptions map to the BORO foundational ontology. This means, for example, determining what terms like activity, role, input and output refer to and how they are modelled in a 4D ontology. Once this mapping is decided it is then possible to proceed with the lexico-semantic analysis aimed at automatically generating a semantic model grounded in BORO from the textual descriptions of the organisational processes.

To identify roles in a business activity it is not only necessary to establish **who** has initiated the activity and **who** is doing the obligatory tasks, but also to establish **what** is the object of the activity (in our examples those are usually documents or sometimes tools, like computers or computer software). From a 4D perspective both subject and object in an activity are treated similarly, because both have spatial-temporal extensions that (in the observed activity) run in parallel and both will change states, triggered by creation or dissolution events. One very common example of business activity is a situation in which a person needs to create or analyse a document (the document could be an input into an activity or it could be created within the activity). Lexically we have the typical S-V-O (subject-verb-object) pattern that could be transformed into an RDF (Resource Framework Description) triple or OWL (Ontology Web Language) ontology snippet with two classes and a relationship between them.

Methods for automated ontology construction found in the literature normally just produce class hierarchies, i.e. they are able to identify classes and taxonomic relationships between them. The method proposed in this paper goes further. Thanks to the foundational ontology, it is possible to identify from the textual descriptions other types of relationships, namely temporal whole-parts quite typical when modelling processes. For example, a role like *AP Manager* is represented in BORO as a temporal part with duration (i.e., a state) of someone who is employed at the company in that position. When the *AP Manager* 'receives a supplier invoice and processes a payment' the activity is itself a temporal part of the role of being an *AP Manager*. Similarly documents can also have temporal parts, for example the states of a supplier invoice may be *unpaid* and *paid*. In BORO, temporal parts (states and events) are fundamental, as described in the previous section; hence types of states (and events) must also be represented

along with their subtyping relationships. Figure 4 shows a partial taxonomy of state types for persons and documents and the temporal part relationships. Figure 5 illustrates with a space-time map how states are modelled. In this case *John* is an employee hired at time t_1 as an *AP Manager* (represented as *State #10*). During the time *John* is in this role he receives *invoice n.100* at time t_3 (the supplier sent it at t_2). From t_3 to t_4 he processes the payment (*state #11* – temporal part of *state #10*). At t_4 he pays the invoice. With this payment *invoice n.100* passes from *state #12* (unpaid) to *state #13* (paid) with both being temporal parts of the invoice. Events are bounding *John*, *Invoice n.100* and all states in correspondence of times t_1 to t_4 . For example, the payment occurs at time t_4 .

Sometimes, however, it may be difficult to distinguish between the state and the event. For example, a common task in a business activity is sending or posting the documents onwards to other entities. In the real world that task will have its place and it will take a reasonable and measurable amount of time. In that particular time stretch a person will be in a state of a *sender* (which includes preparing the documents for sending, using the tool for sending the documents, etc.). Still, from the perspective of a business process, that final task, of the document being sent, takes primary importance. Hence, we intentionally model just that last final action as an event.

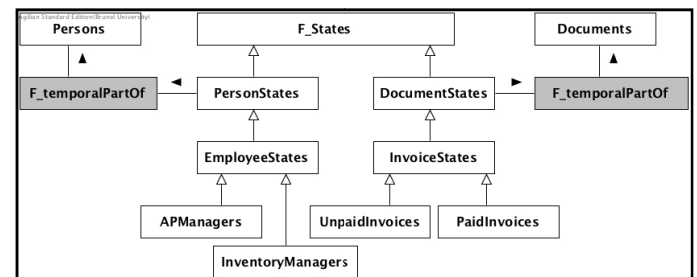


Figure 4. Example of a taxonomy of states and temporal part relations.

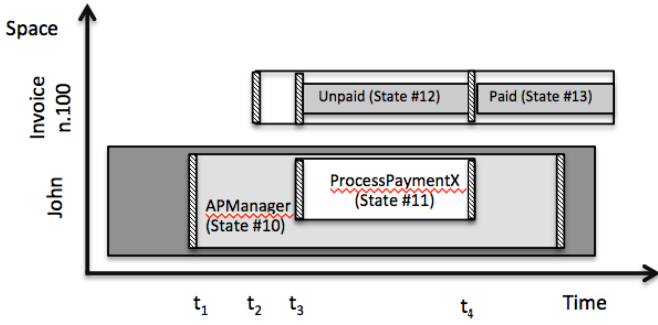


Figure 5. Space-time map of 'invoice payment'.

The method for creating 4D models will always depend on the format of the data we use as an input. Companies usually use some kind of graphical representation of their business processes or just plain text or spreadsheets. Our dataset came in the form of workflow diagrams accompanied by semi-structured text that describes the workflows. The structure of the textual descriptions of business activities is not very rich, but it is repeated in the following pattern: If A is a set of elements describing one particular business process activity then for every A we have $A_N, I_1, \dots, I_n, O_1, \dots, O_k, R$ where N is a string containing a very coarse description of an activity, I_n is a set of strings describing elements that are entering the activity at its beginning. Set O_k is a set of strings describing the entities that are created at the end of the activity, and R represents the string naming the entity in charge of the activity (or role).

The method can be summarised using the following transformations:

- Creation of the class that will represent the activity type. This transformation is straightforward, each activity in the process is named with a substring of N and this transformation can be completely automated (for example, *Source & Enable suppliers* would be a class in the ontology representing the activity type)
- Syntactic and semantic analysis of string N - the string contains the very coarse description of tasks that will be carried out in the activity. It is the only source of knowledge that can give the domain expert an insight into what possible states and events those tasks will bring into existence. An example string from the dataset is "*Pay suppliers, post payments & settle account*". The string is broken into smaller substrings where special characters like comma or & are treated as delimiters (example: $N_1 = \text{"pay suppliers"}$, $N_2 = \text{"post payments"}$ and $N_3 = \text{"settle account"}$). To each word from a particular substring we attach its part of speech (POS) (i.e, Verb (V), Noun (N) or Adjective (A)). We focus on the POS pattern in the form of: $V_1 \dots V_n (A) N_1 \dots N_k$. Each extracted pattern will become a potential class in the 4D ontological model.
- A domain expert then decides on the following actions:
 - Action 1: if candidate classes are semantically meaningful and relevant for the activity, then they should be modelled as states or events.

Example: Pay / Post / Settle \rightarrow events (Figure 6)

- Action 2: the set of candidate classes should be merged (example: description contains synonyms or tasks that imply the same state, like manage and analyse). This is something that should be done in a later stage when an expert decides that some concepts are not necessary and that the generalisation would make the ontology more useful for its purpose.
- Transformation of the role/who entity to the corresponding class in the ontology is straightforward because role is explicitly stated in the data with the interrogative pronoun **who** (representing the role that should carry out all the tasks in the described activity). The new class should represent states that instances of that role type go through while carrying out all required tasks. In most cases role is a temporal part of a person although it can be anything capable of doing the required task for the observed activity (for example, a completely autonomous machine).
- In a 4D ontology activities are represented as mereological sums of all states and events that take part in a specific activity. As an implication of this, for each role involved in an activity there is an element that is being used by the role as a tool necessary for carrying out the activity. Having this in mind we can target these objects for extraction:
 - Roles (usually persons but also software or machines capable of having an impact on a business process) - extract temporal part of a role involved in the observed activity.
 - Events and states - extract the creation event that triggered the activity and the existence of states of both the role and the object of the observed activity.
 - Elements - extract objects of activities together with temporal parts they occupy in the observed activity.

Although these targets can be obtained with manual extraction, for the reasons already explained in the introduction, in the next section we present an algorithm capable of extracting the previously mentioned targets in a completely automated way with the BORO ontology in RDF as an output (as can be seen on Figure 6). For the visual representations we used a modified version of UML class diagrams similarly to those used by the IDEAS group¹. The adaptation was necessary in order to better conform to BORO semantics.

V. AUTOMATIC EXTRACTION OF LABELS

The Propbank database [24] describes relations between predicates and their arguments in the Penn Treebank corpus [25]. Every verb from the corpus is associated with some

¹ International Defence Enterprise Architecture Specification: <http://www.ideasgroup.org/>

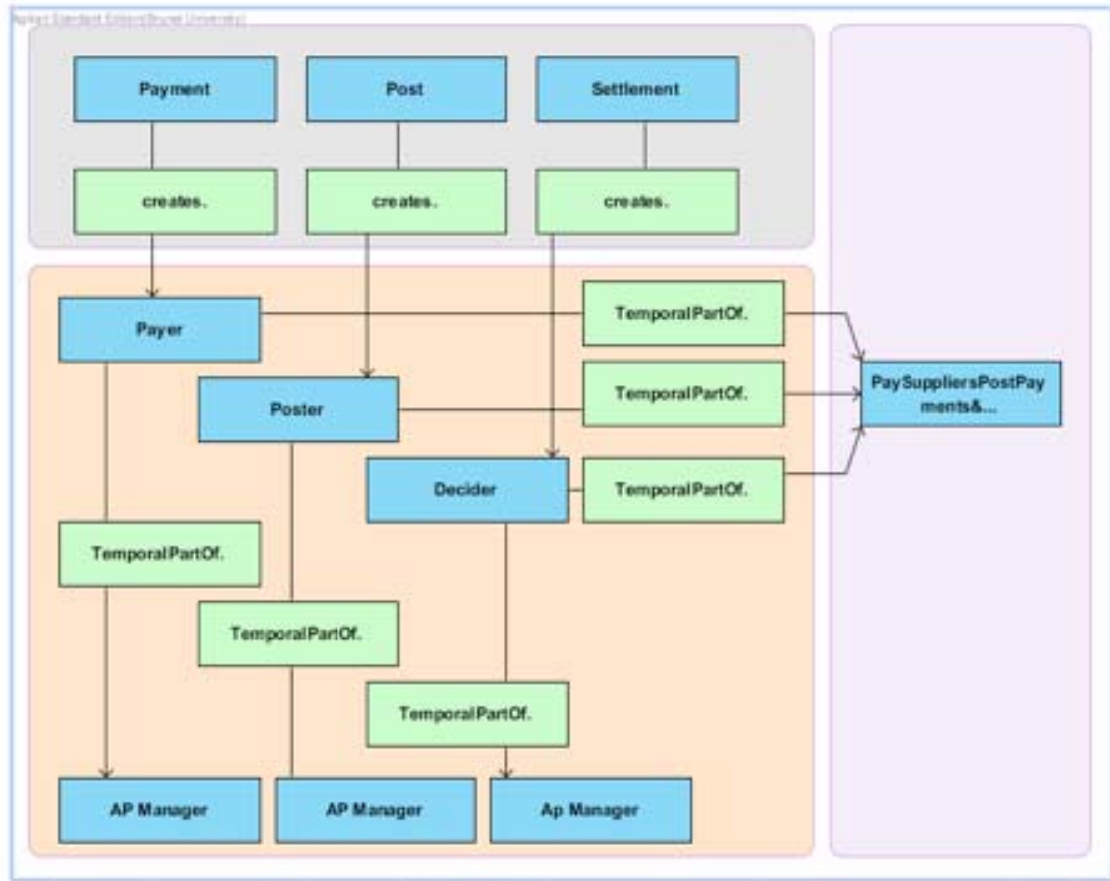


Figure 6. Example ontology: Pay suppliers, post payments & settle account (part).

semantic role, also called argument, which connects the verb with the subject (the agent) and the object (the patient/theme). Each argument is given a number. While some authors differentiate between patient and theme, the Penn Treebank regards it as a single patient/theme argument. The agent and the patient are always given the argument numbers 0 (Arg0) and 1 (Arg1), respectively. In cases where Arg0 does not exist, the role of the agent is given by Arg1. For instance, given the input S-V-O (subject-verb-object) pattern (*buyer receive invoices*), the names for the states of an agent and the patient will be *receiver* and *thing gotten*, respectively. Figure 7 shows how an S-V-O pattern can be broken down into events and states that are caused by the event. Extracted states and events are used as building blocks for the 4D ontology.

Our algorithm for naming the states of the agent and the patient is given an S-V-O pattern as the input (Fig. 8). We apply WordNet to transform the original predicate verb (P) to its infinitive. Next we use the Unified Verb Index (UVI) [26] to find mappings between the verb infinitive and the PropBank entries. UVI is semantic resource available online which merges links and Web pages from four different natural language processing projects including Propbank. If the mapping is successful, appropriate XML files are returned and the values for arguments *Arg0* and *Arg1* are extracted. The values are used for naming the states of both the agent and the patient, which are triggered by a (creation) event (for example,

the verb *receive* will be connected with *Arg0=receiver* and *Arg1=thing gotten*).

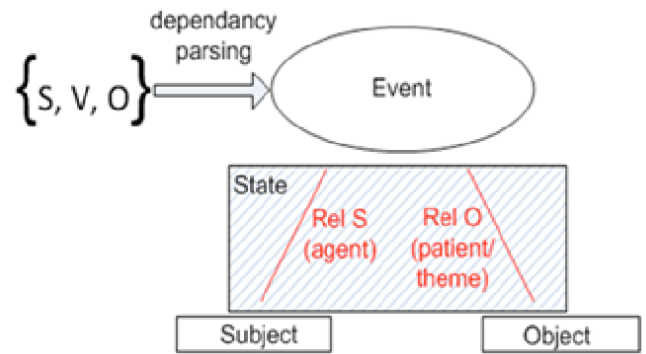


Figure 7. S-V-O patterns as a combination of states and an event.

To extract the appropriate name for the event we convert the verb to its infinitive form and then query WordNet to extract derivationally related forms. Since a large number of different senses of derivationally related forms can be returned, the algorithm extracts the gloss (dictionary definition) for each of the returned senses and analyses them in search for predefined clues (a list of words with meanings similar to words *act* or *event*).

VI. EVALUATION

From 150 business activities in our dataset our algorithm found 177 S-V-O patterns. Table 1 shows the results of the evaluation of the algorithm output (checked manually by a domain expert). Our main concerns were:

- The infinitive form can be obtained.
- The verb exists in the PropBank database.
- The appropriate name for the state of the subject involved in the activity could be extracted.
- The appropriate name for the state of the object involved in the activity could be extracted.
- The appropriate and meaningful derivationally related form that would denote the event could be extracted.

For 70.62% of the patterns the algorithm was successful in obtaining meaningful names for the states of the subject and the object involved in the activity together with the name of the event (this case received a grade 2). The word meaningful here means that the extracted labels can be used inside the ontology without changing the meaning that the terms would make readers infer from the initial description; for example, the description *allocate invoices* will produce an event called *allocation* and arguments with labels *granter* and *thing granted*, which are all correct in a given context. In 28.24% of the cases meaningful names for the states of subject and object were successfully extracted but the name of the event was non-existent. Those cases were not excluded from the ontology but the verb was used for naming the event (we have to stress that this solution, although not perfect in some cases, still gives us reasonably meaningful names for the class that will represent the event in the 4D ontology). In just two cases (1.12% percent) the algorithm produced names that were completely wrong and not suitable for the ontology (both cases came from the same verb, *execute*, with the arguments *killer* and *corpse*).

TABLE I. EVALUATION RESULTS

Grade	2	1	X
Number	125	50	2
Percentage	70.62	28.24	1.12

Of 216 sentences (of which 177 contained the right pattern) 37 (or 17.12%) were excluded at the beginning of the algorithm because the right pattern could not be identified. Although, that number is not considerably large, evaluation of those cases pointed to problems with parsing (imperatives often used in workflows are especially problematic).

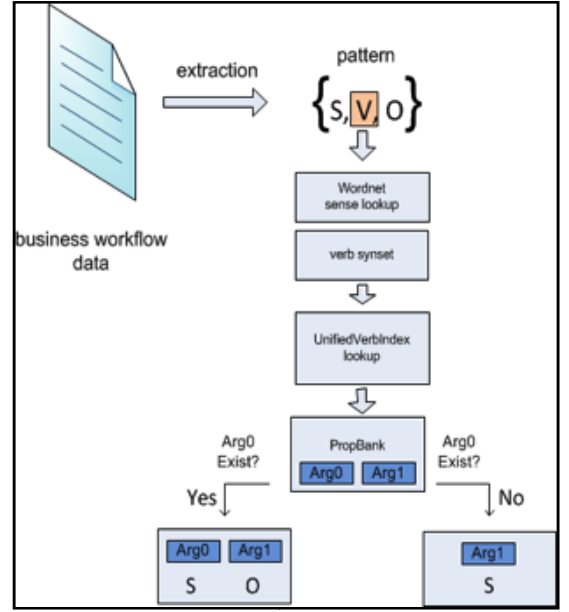


Figure 8. Algorithm for naming events and states of the agent and the patient.

This evaluation showed that semantic roles and derivationally related forms can be mapped to 4D ontology objects and that workflow descriptions can be used as a valuable source for linguistic analysis and ontology construction. In comparison with similar efforts of ontology construction from raw or semi-structured texts like Wikipedia articles, workflows offer a source of data that is very compact and concise and therefore not so prone to mistakes that can come into existence as a consequence of analysis of highly complex textual sources.

The algorithm used semantic repositories such as WordNet [27] and PropBank [24], and the Stanford dependency parser [28]. The final step of our method was the creation of a semantic repository populated with triples that describe the events and states previously extracted from the business process workflows together with roles and temporal relationships. The creation of the semantic repository allows us to query the original data in ways that were not possible before (for example, how many states does one person in a process have to go through before the activity comes to an end?). Table 2 shows the total number of triples created from the dataset, number of entities in the repository and number of temporal parts, creation events and distinct roles extracted by our method.

TABLE II. REPOSITORY STATISTICS

Total nr. of triples	nr. of entities	nr. of creation events	nr. of temporal parts	nr. of distinct roles
745	230	100	414	19

RDF is used to store the created ontologies (an example is provided in Figure 9).

<pre> <rdf:Description rdf:about="http://purl.org/lembo#thing approved"> <rdf:type rdf:resource="http://purl.org/lembo#State"/> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#requestion"/> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#advance"/> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#request"/> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#timesheet"/> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#expenses"/> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#authorisation"/> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#proposal"/> </rdf:Description> </pre>	<pre> <rdf:Description rdf:about="http://purl.org/lembo#separation"> <embo:creates rdf:resource="http://purl.org/lembo#thing being divided"/> <embo:creates rdf:resource="http://purl.org/lembo#entity causing separation"/> <rdf:type rdf:resource="http://purl.org/lembo#Event"/> </rdf:Description> </pre>
	<pre> <rdf:Description rdf:about="http://purl.org/lembo#Requestor"> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#Person"/> <rdf:type rdf:resource="http://purl.org/lembo#Role"/> </rdf:Description> </pre>
	<pre> <rdf:Description rdf:about="http://purl.org/lembo#Activity2"> <embo:temporalPartOf rdf:resource="http://purl.org/lembo#Process5"/> <rdf:type rdf:resource="http://purl.org/lembo#Activity"/> </rdf:Description> </pre>

Figure 9. RDF representation of business activity

VII. CONCLUSION AND FUTURE WORK

Although the construction of business ontologies by ontologists and domain experts will always be necessary due to the sensitivity of this kind of data on the business, we think that this cumbersome and time demanding process should not be completely manual. This paper is a step toward automating such a process. We developed a method capable of analysing workflow descriptions of business processes and their automatic transformation into a BORO-based ontology, by using various computer linguistic methods and openly available semantic resources. The created ontology elements were automatically transformed into RDF elements and stored into a semantic repository that allows us to query for interesting facts from a semantically rich model that describes the business processes. The developed method enables a more efficient production of the core parts of the ontology and providing the business ontologist with a more advanced starting point at the very least. Our method was designed to work with textual workflow data and the BORO ontology was chosen because it allows for more accurate semantic representations of socially constructed objects like processes, which are quite common to a business environment. Since the method is intended to assist business analysts, we are currently working on a user-friendly tool that will visualise the data and models automatically extracted with the method. The tool would present business process elements in a BORO representation and enable analysts/ontologists to edit, change and finally store the new patterns into the semantic repository with an endpoint for querying the stored data.

REFERENCES

- [1] Hepp M., Roman D.: An Ontology Framework for Semantic Business Process Management, Proceedings of Wirtschaftsinformatik 2007, 28.02 - 02.03, 2007, Karlsruhe
- [2] Partridge, C. 1996, Business Objects ReEngineering For ReUse, 1st edition edn, Butterworth-Heinemann College.
- [3] Sider, T. 2003, Four-Dimensionalism: An Ontology of Persistence and Time, Oxford.
- [4] de Cesare S., Geerts G.L.: Toward a Perdurantist Ontology of Contracts, Advanced Information Systems Engineering Workshops, Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, 2012.

- [5] Curtis J, Cabral J, Baxter D. On the application of the Cyc ontology to word sense disambiguation. In: FLAIRS Conference. AAAI Press; 2006, p. 652–7.
- [6] Fellbaum C, editor. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press; 1998.
- [7] Niles I, Pease A. Towards a standard upper ontology. In: FOIS. 2001, p. 2–9.
- [8] Krotzsch M, Vrandečić D, Volkel M, Haller H, Studer R. Semantic Wikipedia. Journal of Web Semantics 2007;5(4):251–61.
- [9] Zhou L. Ontology learning: state of the art and open issues. Information Technology and Management 2007;8(3):241–52.
- [10] Wisniewski M. Metamodel of ontology learning from text. In: Emergent Web Intelligence: Advanced Semantic Technologies. Advanced Information and Knowledge Processing; Springer; 2010, p. 245–76.
- [11] Agichtein E, Gravano L, Pavel J, Sokolova V, Voskoboynik A. Snowball: A prototype system for extracting relations from large text collections. In: SIGMOD Conference. 2001, p. 612.
- [12] Snow R, Jurafsky D, Ng AY. Semantic taxonomy induction from heterogeneous evidence. In: ACL. 2006, p. 801–8.
- [13] Cunningham H, Maynard D, Bontcheva K, Tablan V. A framework and graphical development environment for robust NLP tools and applications. In: ACL. 2002, p. 168–75.
- [14] Cimiano P, Hartung M, Ratsch E. Finding the appropriate generalization level for binary ontological relations extracted from the GENIA corpus. In: LREC. 2006, p. 161–9.
- [15] Pantel P, Pennacchiotti M.: Automatically harvesting and ontologizing semantic relations. In: Ontology Learning and Population: Bridging the Gap between Text and Knowledge; 2008, p. 171–95.
- [16] Maedche, A., Staab, S., The Text-To-Onto Ontology Learning Environment. In: Software Demonstration at the Eighth International Conference on Conceptual Structures (Springer-Verlag, Berlin, 2000).
- [17] ConceptNet5: <http://conceptnet5.media.mit.edu/>
- [18] A De Nicola, M Missikoff, R Navigli, A software engineering approach to ontology building; Information Systems 34 (2), 258-275
- [19] D'Antonio F., Missikoff M., Taglino F.: Formalizing the OPAL eBusiness ontology design patterns with OWL. Enterprise Interoperability II, pp 345-356, Springer London, 2007.
- [20] Thomas O., Fellmann M.: Semantic epc: Enhancing process modeling using ontology languages. In Proc. of the Workshop on Semantic Business Process and Product Lifecycle Management at the ESWC, volume 251 of CEUR-WS, 2007.
- [21] van der Aalst, WMP, ter Hofstede, A., Kiepuszewski, B. & Barros, A. 2003, "Workflow Patterns", *Distributed and Parallel Databases*, vol. 14, no. 1, pp.5-51.
- [22] Thom, L.H., Iochpe, C. & Reichert, M. 2007, "Workflow Patterns for Business Process Modeling", Proceedings of Workshops and Doctoral Consortium of the 19th International Conference on Advanced Information Systems Engineering (CAiSE), vol. 1, pp. 349-358.
- [23] Aldin, L. and de Cesare, S. A Literature Review on Business Process Modelling: New Frontiers of Reusability. Enterprise Information Systems, 5(3), 2011, pp. 359-383.
- [24] Palmer M, Gildea D, Kingsbury P. The Proposition Bank: An annotated corpus of semantic roles. Computational Linguistics 2005;31(1):71–106
- [25] Penn Treebank project. <http://www.cis.upenn.edu/~treebank/>
- [26] Unified Verb Index. <http://verbs.colorado.edu/verb-index/>
- [27] Fellbaum C. (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.
- [28] de Marneffe, C.-M., MacCartney, B., Manning, C.D.: Generating Typed Dependency Parses from Phrase Structure Parses. Proc. LREC 2006, 449-454.