Automated taxonomy extraction from semantic business process models

Sergio de Cesare, Damir Juric and Mark Lycett
Brunel University London, U.K.
{sergio.decesare|damir.juric|mark.lycett}@brunel.ac.uk

Abstract

Business process patterns, or generalised reusable process models, have the potential to reduce the effort and costs of developing new business processes or re-engineering existing ones. Identifying new patterns, however, requires an analysis of existing business data to uncover commonalities of processes within and across organisations. In previous work we developed a method to extract linguistic content from semi-structured data and generate a semantic repository of such content in a graph database. The method adopted a foundational ontology to semantically underpin the developed graph models. This paper is aimed at identifying commonalities within the process models stored in the semantic repository. We adopt a linguistic approach based on semantic similarity in order to compare the least common subsumers (LCS) of pairs of nodes in the semantic graph. Each LCS is then used to generate a generalised (supertype) node in the graph. The method is presented and evaluated.

1. Introduction

Business Process Management (BPM) is being increasingly adopted in modern organisations as a systematic way of representing, monitoring and improving their processes [1]. Though BPM has had considerable take-up in industry, the current state-of-the-art still leaves organisations with both limited knowledge of their process space and incomplete and delayed control [2]. This situation arises due to the need for manual arbitration between the business perspective on operations and the execution of those operations in information technology (IT) systems. Such arbitration is required in both getting operational data from IT systems and, importantly, in enacting and changing processes in (increasingly) dynamic business environments.

The next evolutionary step requires the process space to be continuously managed and integrated. Envisioning BPM in terms of semantic repositories arguably paves the way for greater automation and knowledge of the process data space. Both are important in dynamic environments, as it is their combination (often via analytics) that enables ‘value’ [3]. Significant challenges exist, however, in making sense of data in an increasingly ‘datafied’ process space [4]. Data related to the process space is increasing in volume, velocity, variety and veracity. Variety alone, for example, presents data in structured, semi-structured or unstructured form, progressively providing more challenges in automatically interpreting and translating into a form that can be processed by software systems. Significant challenge is also presented in conceptualising and codifying the process space if the maximum ‘value’ from such data is to be realised in developing, enacting and changing business processes.

As part of a continuing line of work [5] we are approaching such problems via the development of a methodology for the (semi-automated) discovery of business process patterns from organisational data sources. A business process pattern, in this context, is defined as a generalised semantic model that can be reused (either as it stands or by specialising it) in other BPM projects (either within/across organisations or within/across domains). The methodology is: (1) Empirically based, since patterns are discovered from organisational data sources; and (2) semantic, since all models are grounded in a foundational ontology (i.e. BORO [6]). In developing our work here, we use the semantic representations to discover similarity among different models to develop a taxonomy of business process representations at successive levels of generalisation (i.e. generalised patterns). This implies also discovering taxonomies of business process elements such as activities, events, roles and resources. We then integrate the taxonomies into a semantic graph repository (graph database). Our approach is to compare business process data stored in the graph database repository and use semantic similarity measures to determine whether two or more existing types in the ontological model can be subsumed by a more general type. In other words the underlying algorithm aims to calculate the Least Common Subsumer (LCS).

In achieving our aims, the paper is structured as follows. Sections 2 and 3 introduce the BORO foundational ontology, reasons for its adoption and an understanding of how business processes are represented in BORO. Sections 4 and 5 describe the problem of comparing business process models in a graph repository. These sections propose a solution for discovering more general types that subsume existing ones, thus expanding the semantic repository. Section 6 describes the evaluation of the proposed solution.
Finally Section 7 presents related work and conclusions.

2. The BORO Foundational Ontology

This research adopts the Business Object Reference Ontology (BORO) [6] to semantically interpret the original organisational datasets and models. We adopt BORO as it provides a means to overcome 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.). BORO represents all individual elements (e.g. the activity, the person assuming a role and the resource consumed) in exactly the same way (i.e. as spatiotemporal extents).

BORO is based on a philosophical (rather than computational) definition of ontology because it requires more clarity on “the set of things whose existence is acknowledged by a particular theory or system of thought” [7]. Key to overcoming the dichotomy noted is the fact that BORO is perdurantist and extensionalist. In perdurantism (or 4D) an individual object is never wholly present at one point is time, but only partly present (a temporal part). For example, John is not fully present in any given phase of his life (e.g., childhood), he is fully present from his birth to his death only – John’s childhood is a temporal part of John. Identity is therefore defined by an individual object’s spatiotemporal extension (or extent).

In turn, Elements is subtyped by:

- **Events**: An event is an element that does not persist through time (i.e. an event has zero ‘thickness’ along the time dimension). Events represent temporal boundaries that either create (CreationEvents) or dissolve (DissolutionEvents) elements (e.g., a person or a person’s childhood state).

- **States**: A state is an element that persists through time. States (and elements in general) are bounded by events. A state (like all elements) can have further temporal parts (i.e. states and events).

Specific **TupleTypes** (or types whose instance are tuples) relevant here are:

- **temporalPartOf**: This tuple type relates an event with one or more elements affected by the event. temporalPartOf has two subtypes:
  - **creates**: Relates a creation event with the element(s) whose creation is triggered by the event.
  - **dissolves**: Relates a dissolution event with the element(s) whose dissolution is triggered by the event.

- **happensIn**: This tuple type relates an event with a time instant or interval (TimeInstantsOrIntervals) and it indicates the time in which an event takes place.

- **mereologicalSums** (or fusions): This tuple type relates a type (i.e. set of elements) with its fused whole. In other words, all instances of the former are parts of the latter and represent a complete cover of the whole. For example, John’s childhood, teenagehood and adulthood form a set whose fusion is John. All three elements (stages) are temporal...

![Figure 1. BORO foundational ontology (partial view)](image-url)
parts of John’s life and completely cover John as a 4D extent.

3. Business Processes in BORO

A comparative analysis of the extant business process modelling literature [8] shows the key modelling concepts to be:

- **Business process**: A set of activities, events, etc. that together cohesively delivers a service and/or a product.
- **Business activity**: Specific behaviour carried out in an organisation.
- **Business event**: An occurrence that takes place at a specific point in time and that is capable of triggering or terminating some observable behaviour (activity or process).
- **Business role**: The types of actors or agents that take part in processes.
- **Goal**: The intention for which the process was designed.

From a semantic perspective these business process constructs need to be better grounded in reality which, in the case here, means interpreting them in light of the 4D foundational ontology and the ontological choices that it makes. For sake of brevity we do not present the complete semantic analysis, but just its outcome. In summary:

- A business process is the fusion (or mereological sum) of all temporal parts that together create things (with 4D extent) to satisfy the requirements/needs of an internal (e.g., other organizational unit) or external (e.g., customer) person (human or organisation). A business process differs from other fusions (e.g., the fusion of John’s childhood, teenagehood and adulthood) because it: (1) Is always socially constructed; (2) exists in relation to an organisation; (3) has an intended goal modelled as a 4D extent.
- A goal may be the production of a new thing (e.g., a car) or change in an existing thing (e.g., person attending a theatrical performance, which equates to a change of state in that person or new temporal part). In both examples, however, a new 4D extent is always created.
- A business activity is also a fusion of temporal parts. Unlike a process, an activity does not, on its own, completely satisfy the requirements or needs of an internal or external person, it does so in conjunction with other activities. A process is the fusion of zero or more activities: An activity can be composed of other activities.
- A business role is a temporal part of an element (person, machine, software, etc.) that is capable of being responsible (either directly or indirectly via delegation) of an activity or a process. For example, while a person is testing electrical components in a car production line, there is a spatiotemporal extent of that person which instantiates the role type ComponentTesters.
- A business event is a BORO event that is a temporal part of a business process and can initiate or terminate a process, an activity or a role.

4. Semantic Similarity of Business Process Models

This paper builds upon the research presented in [5]. There we presented a method for the automated transformation of semi-structured business data into ontological models. The algorithms proposed in [5] and in this paper are computationally driven by Natural Language Processing (NLP) and semantically underpinned by the BORO foundational ontology. Relevant objects from textual descriptions of business processes were extracted and semantically interpreted. These objects included, for example, the events, activities, roles and resources taking part in processes. The elements extracted from the original models were linguistically analysed and lexically classified according to subject-verb-object (SVO) and patient-agent linguistic patterns: These patterns were then mapped to BORO foundational patterns and subsequently stored in a graph repository. This phase of research concentrates on identifying commonalities that can be generalised within the semantic graph.

In moving to the problem at hand, there are three possible approaches to compute similarity between business process models [9]: Syntactic, linguistic and structural. Syntactic similarity measures simply check the number of identical characters that appear in model labels. Linguistic measures consider possible synonyms for each model label, which can be queried from linguistic databases like WordNet. Structural similarity measures consider the placement of a label inside the model and consider its context. The method described in this paper adopts all three approaches and adapts them in the following manner. Business process descriptions are not compared in their original form, but are syntactically and linguistically analysed as described in [5]. The algorithm used produces a set of subject-verb-object patterns in which the subject assumes the linguistic role of an agent and the object that of a patient; the agent acts upon the patient. For example, in the case of component testing, the tester tests testedThing. The tester (agent) acts upon the testedThing (patient). This linguistic pattern is semantically transformed to conform to BORO. Since roles are modelled in 4D as temporal parts (BORO
the algorithm automatically creates the two states of Tester and TestedThing, with the former as a temporal part of an employee and the latter as a temporal part of the component. Since all Elements are created by CreationEvents, the relative events are also generated as the starting temporal boundaries of Tester and TestedThing. This is illustrated in Figure 2. Moreover, a ComponentTesting activity is also created as the mereological sum of the Tester and the TestedThing. As the ontological model grows, the nodes and edges in the semantic graph repository are added. Though an aside here, issues specific to the transformation of BORO models to a graph are presented in [10].

Figure 2. Mapping between linguistic elements and BORO objects

This algorithm appropriately addresses the syntactic and linguistic issues and, via the foundational ontology, creates models that conform to the same structure: The problem of generalising (or finding commonalities) across business process models still remains however.

Figure 3 shows two business process graph patterns. Both patterns have unique label identifiers for process and activity type nodes. Below the activity type nodes are event type nodes. If two patterns share the same node then there is evidence of a possible commonality, which may lead to a generalised pattern. Alternatively the process patterns may contain different nodes which are semantically similar. This scenario is presented at the bottom of Figure 3; here similarity between the nodes at the bottom right of each pattern can be calculated and a more general node can be produced. The new node (illustrated in Figure 3 with a question mark) represents a more general concept and it is called the Least Common Subsumer (LCS). For example, events such as checking and reviewing can be categorised under the general concept of inspection, while nodes such as expense_information and receipt_claim can be subsumed by the more general node message.

Figure 3. Relations between two process patterns in the semantic repository

5. Generalisation Method

Figure 4 shows a section of the semantic graph described above (screenshot from the Neo4J graph database). The central node of the graph is thingChecked. Node thingChecked was created after performing a linguistic analysis of business process descriptions. This node is connected via temporalpartof relationships to other nodes represented as BORO elements (e.g., expense_claim, duplicate, invoice, services, goods, materials, etc.). These nodes also appeared after linguistic analysis of business process activity description (all those terms were found in activities descriptions related to the act of checking). The algorithm presented in this section lexically analyses the names of two elements at a time in order to find a term in WordNet [11], which represents a common generalised term, whose linguistic semantics is common to the two original terms.
In light of the above the following can be defined:

- For some $EV_{BORO}$ types (events) there is a semantically appropriate label that can be used when a derivation for $V_{infinitive}$ (verb) can be found in WordNet [10], $V_{drf}$ (i.e., terms in different syntactic categories that have the same root form and are semantically related).
- For every $EV_{BORO}$ there is a triplet $T'_{BORO}=$\{$EV_{BORO}$, $CR$, $ST'_{BORO}$\} where $CR$ represents a relationship (labelled $creates$) and $ST'_{BORO}$ is a BORO state (labelled with a suitable agent semantic role) and $T''_{BORO}=$\{$EV_{BORO}$, $CR$, $ST''_{BORO}$\} where $ST''_{BORO}$ is a BORO state labelled with a suitable patient semantic role or triplet $T'''_{BORO}=$\{$EV_{BORO}$, $CR$, $ST'_{BORO}$\} when only an agent semantic role exists (for a given event $EV_{BORO}$).
- For every $ST'_{BORO}$ and $ST''_{BORO}$ there is a triplet $T'_{BORO}=$\{$ST'_{BORO}$, $TR$, $STs_{BORO}$\} and $T''_{BORO}=$\{$ST''_{BORO}$, $TR$, $STo_{BORO}$\} in which $TR$ represents a relationship (labelled $temporalPartOf$) and $STs_{BORO}$ and $STo_{BORO}$ are BORO states (labelled with suitable agent semantic role).

Labels in business process data usually contain a fairly limited number of verbs in the textual representations referring to activities: This is due to the simple fact that the number of nouns in language is far greater than the number of verbs. Processes across different domains are textually represented with a vocabulary of activities like read, check, send and analyse. As noted earlier, however, the individual elements that those activities involve can vary greatly across domains. An important step toward comparing and generalising/specialising processes consists of categorising the things that those processes/activities handle, manipulate, manage and so on. In BORO semantics these temporal parts of these individual things would fuse to form the process or the activity.

The algorithm we have developed aims at linguistically analysing the labels of these elements in order to identify the LCS. This algorithm is described below and illustrated at Figure 5.

![Figure 5. Algorithm flow](image-url)
For each element of set PS the graph is traversed for a BORO pattern PatientState\_temporalPartOf \_object like thing\_checked \_temporalPartOf \_invoice


- Found objects are considered as candidates for similarity calculation after they are being stemmed \{\textit{information}, \textit{claim}, \textit{duplicate}, \textit{accuracy}, \textit{invoice}, \textit{detail}, \textit{service}, \textit{good}, \textit{material}, \textit{detail}\}. Also another set similar to this one is created and populated with event labels.

- Each object is paired with another object from the set so we have a set of pairs \{(\textit{information \_claim}), (\textit{information, duplicate}), ..., (\textit{good, material}), ..., (\textit{material, detail})\}.

- Semantic similarity between two elements of a pair is calculated. Since words can have different meanings (senses in WordNet), we consider every possible meaning for the word, however we are only interested in senses that produce maximum similarity. For example:

  Similarity (\textit{information} \#n\#\{1-5\}, \textit{claim} \#n\#\{1-6\})

- When the maximum similarity between two elements (their particular senses in WordNet) is found their corresponding hypertrees are analysed \((HT_1 \text{ and } HT_2)\). For example:

  \[
  HT_1(\text{information\#n\#1}) = *\text{ROOT}^*\#n\#1 < \text{entity}\#n\#1 < \text{abstraction}\#n\#6 < \text{communication}\#n\#2 < \text{message}\#n\#2 < \text{information}\#n\#1
  \]

  \[
  HT_2(\text{claim\#n\#6}) = *\text{ROOT}^*\#n\#1 < \text{entity}\#n\#1 < \text{abstraction}\#n\#6 < \text{communication}\#n\#2 < \text{message}\#n\#2 < \text{request}\#n\#1 < \text{demand}\#n\#1 < \text{claim}\#n\#6
  \]

  \[
  \text{argmax}(\text{depth(subsumer}(HT_1, HT_2))) = \{\text{message}\#n\#2\}
  \]

- At the intersection of hypertrees \((HT_1 \cap HT_2)\) lies a Least Common Subsumer that represents the concept having the minimal number of edges between the LCS and its two hyponyms in WordNet. For example:

  \[
  \text{Least Common Subsumer(s)} = \text{argmax}(\text{depth(subsumer}(HT_1, HT_2))) = \{\text{message}\#n\#2\}
  \]

- If the LCS exists a new node is created in the ontological model and a new \textit{subtype} relation between the newly created node (derived from the LCS) and elements of the object pairs is created.

- The final result is the addition of two new relations in the ontological graph for each similarity calculation that is not an empty set:

\[
T_{\text{boro}} = \{\text{ST}_{Oi}, \text{TP}, \text{ST}'_{LCS}\} \text{ and } T'_{\text{boro}} = \{\text{ST}_{Oj}, \text{TP}, \text{ST}'_{LCS}\}.
\]

For example:

\[
T_{\text{boro}} = \{\text{receipt\_information, temporalPartOf, message}\}
\]

\[
T'_{\text{boro}} = \{\text{expense\_claim, temporalPartOf, message}\}
\]

Pseudocode for the algorithm can be seen in Figure 6.

![Figure 6. Pseudo code of the algorithm](image)

Some of the discovered nodes can be seen on Figure 7. The algorithm calculates similarity between the names of two BORO objects by using a semantic similarity measure based on WordNet. Least Common Subsumers (more general concepts of two BORO objects for which we measured semantic similarity) can be seen in Figure 7 as nodes that are now incorporated into the graph database. Though several semantic similarity measures exist, they need to include the LCS in their formulas. This narrows the candidate set down [see 12, 13 and 14 for example]. Our early experiments indicate that there is little difference between these approaches so, for the time being we have adopted [12]. Here the approach calculates relatedness by considering the depths of the two synsets in the WordNet taxonomies, along with the depth of the LCS. The formula is:

\[
\text{score} = 2 \times \text{depth(lcs) / (depth(s1) + depth(s2))}
\]

The implication of the formula is that the score can never be zero because the depth of the LCS is never zero (the depth of the root of a taxonomy is one). The score is one if the two input terms are the same.
6. Evaluation

The effective evaluation of ontology remains an open research question. Here, we have approached the task using semiotic theory [14], which adopts the following evaluation criteria:

- **Syntactic quality** – is the ontology readable (the way in which the ontology is written)?
- **Semantic quality** – can the ontology be understood (meaning of signs, what they represent)?
- **Pragmatic quality** – for what will the ontology be used and can it give answers to proposed questions?
- **Social quality** – potential and social consequences of the ontology.

The source data used to evaluate our algorithm is a business process ontology describing procurement processes. The ontology was created automatically from textual descriptions of 29 business processes that contained 150 business activities. These processes were provided by an industrial partner.

From a syntactic perspective, the algorithm presented in this paper automates the process of creating taxonomies inside an already existing business process ontology and incorporates newly discovered elements (nodes) to a graph database. We can therefore say that the ontology is readable and its syntactic quality is guaranteed in the coding process (formalized ontology is an output of an ontology building algorithm). Quality, in this sense, is guaranteed via accordance with the BORO foundational ontology and the operation of the graph transformation algorithm.

From a semantic perspective the measure employed for evaluation is **interpretability** (meaningfulness of terms) [15]. Table 1 shows that 70.62% of the node labels were considered meaningful in our source ontology. By that, we mean that the extracted labels can be used within the ontology without changing the meaning that readers would infer from the initial description. For example, the description *allocate invoices* produces an event called *allocation* and arguments with labels *granter* and *thing granted*, which are appropriate in the given context. Evaluation was carried out by a domain expert who is familiar with the source data (company data) and who is able to decide on the meaningfulness of new terms in the context of company data.

In 28.24% of the cases partially meaningful labels were extracted (in all cases, however, verbs describing activities in a process were problematic). Those cases were not excluded from the ontology, but the verb was used for naming the event: We note that this solution, although problematic in some cases, still gives us reasonably meaningful labels for the class that will represent the event in the 4D ontology. In only 1.12%

![Figure 7. Generated super classes in graph database (part)](image-url)
of the cases (two in number) did the algorithm produce labels that were completely unsuitable for the ontology (both cases came from the same verb, execute, with the arguments killer and corpse).

**TABLE 1. Interpretability of source ontology.**

<table>
<thead>
<tr>
<th>Grade</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>125</td>
<td>70.62</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>28.24</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
<td>1.12</td>
</tr>
</tbody>
</table>

Our algorithm produced nodes as described in section 5. Nodes are measured by their interpretability (meaningfulness). We used the grading system to evaluate meaningfulness of the taxonomic relationship between newly discovered more general nodes with nodes from the source ontology. Grade 2 or 1 is given to the relationships that are meaningful (grade 2 means that the evaluator completely agrees with the relationship and grade 1 means that relationship is not wrong, but the context could be expressed better or a more appropriate general node could be added instead). Grade 0 is given to completely wrong relationships while X is given to pairs of nodes that do not have any similarity, thus there is no general concept that subsumes them. For example: grade 2 for cost (general node) that connects with expense (source node) and payment (source node) with subtype relationship, grade 1 for node human_action that connects to nodes advance and requisition, grade 0 for node speech_act that connects with nodes proposal and request, and grade X for pair of nodes without LCS like timesheet and supplier. Examples of grades 1 give to event level nodes were rare (like bread_and_butter for maintenance/support or chess_move for capture/check). Table 2 shows the number of cases that the particular grade was assigned to node pairs from our test dataset. The first number refers to the linguistic object level and second number to the BORO semantic level (with a focus on events).

**Table 2. Evaluation results (interpretability)**

<table>
<thead>
<tr>
<th>Grade</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>232/1383</td>
<td>54/753</td>
</tr>
<tr>
<td>1</td>
<td>33/14</td>
<td>8.5/0.43</td>
</tr>
<tr>
<td>X</td>
<td>69/1090</td>
<td>17.78/33.64</td>
</tr>
</tbody>
</table>

In pragmatic terms, the measure employed for evaluation is **comprehensiveness** (number of classes and properties). Table 3 (upper part) shows the size of the source ontology of business processes (as the number of nodes and total number of triplets), while the lower part of Table 3 is showing the number of new nodes and triplets created with the algorithm described in this paper. The source ontology is enriched with new nodes that describe more general concepts.

**Table 3. Evaluation results (comprehensiveness).**

<table>
<thead>
<tr>
<th>Source Ontology</th>
<th>Total nr. of triplets</th>
<th>nr. of nodes</th>
<th>nr. of creation rel.</th>
<th>nr. of temporal rel.</th>
<th>nr. of activities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discovered taxonomy nodes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An ontology is relevant if its representation can provide appropriate answers to previously defined queries regarding, for example, activities. One competency question that the proposed method must be capable of answering is:

Q: To which general concepts can the ontological business process patterns be mapped to?

Figure 8 shows the list of distinct anchor nodes that are created by our algorithm from the test data. Nodes of the linguistic object level and BORO (event) level from our repository can be mapped to more general concepts as shown in the figure. Those general concepts are LCS terms discovered by the algorithm.

**7. Related and Future Work**

Though the focus of this paper is not on similarity measures, we are mindful that a corpus of research exists in this area. Approaches to measuring syntactic and semantic similarity between business process models, particularly Petri net-based ones, can be found in [9]. In [16] the authors provided a comprehensive survey on techniques to define and calculate similarity measures for business processes. In this paper authors examined existing natural language processing techniques for their adaptability in the reuse of software models and suggest methods for grouping existing models according to their functionality. Here, however, similarity measures were used as a tool used for achieving the goal of adding additional semantics (taxonomies). We will examine the efficacy and effects of different measures in future work.

Manual ontology development [e.g., 17, 18, 19] and evolution can be slow and others have also sought to move toward automated approaches. Zhou [20], for example, proposes a typical scenario for an ontology learning process (which can either be manual or automated) consisting of concept creation, relationship creation, ontology population and ontology evaluation. Wiszniewski [21] introduces a metamodel for ontology learning from text and presents an extensive survey of ontology learning models. Three different approaches for extracting possible binary relations from text sources have been developed, each focussing on
different features of the source text - collocations, syntactic dependencies or lexico-syntactic patterns respectively. Many approaches in the literature are a combination of these approaches [22, 23, 24].

| [action, knowledge, statement, occurrence, substance, entity, abstract entity, communication, information, psychological feature, human action, activity, artefact, quality, part, grouping, organization, state, possession, legal document, instruction, order, magnitude, papers, cost, event, pass on, commercial instrument, writing, written communication, relation, asking, attribute, position, offer, proposal, speech act, whole, process, scil group, declre, change, arrogate, quest, postulation, individual, condition, control, program] |

From a methodological perspective, authors in [25] adopt a task-based approach (TBM) for developing business process models. TBM defines key verbs in business processes as the basic task components. By supplementing a verb with a performing method and detailed information, its action can be fully defined and its outcome can be measured. Using task components as the basic modelling elements, a business process model can be created by connecting the required activities. This approach is similar to ours in that it tries to identify tasks/activities from verbs (a process that can be automated). TBM still requires a user to identify the key verbs that are essential for the given business in order to use the approach. Our method has potential advantage in that it is built on a tool that uses deep parsing for identification of linguistic patterns (not just verbs, but objects and object dependencies).

In [26] the authors investigate so-called action patterns that capture the fragments of action that often appears together in business processes. The same authors provide insights into activity labelling styles in business process models in [27]. While their approach presents a high level of automation in business process model analysis, their main goal is to identify commonalities or similar actions by using association rules.

Last, the approach described by De Nicola et al. [28] deals with the specific domain of business processes. Their UPON (Unified Process for ONtology), describes phases, steps, and intermediate outcomes to guide ontology engineers in the production of domain ontologies. The work is methodologically focussed and the work is done manually (especially the cumbersome process of storyboard analysis and vocabulary creation). A continuation of this work is found in [29] where, in order to support business experts in building ontologies, a limited number of high-level conceptual templates are presented.

Our future work is aimed at overcoming limitations of the research carried out to date as well as integrate the method presented above within an overarching methodology for the discovery and use of business process patterns in the wider enterprise engineering context.

In relation to the former, while our approach as demonstrated above is capable of integrating linguistic analysis techniques with a foundational ontology, some conceptual inconsistencies need to be resolved. For example, as Figure 7 shows some generalised types include abstract_entity and information (among others). In a realist ontology like BORO all individual objects (elements) are physical with a 4D extent. This means that, unlike those foundational ontologies that take an idealist stance, there is no commitment to abstract objects/entities.

In relation to the latter our ultimate aim is to deliver a methodology that can be used by business process engineers to (re-)develop new or existing processes from previously identified and well documented patterns. This work, along with [5], provide algorithms that will assist in the semi-automated discovery of patterns to be used in conjunction and harmonised with manual business process design activities.

8. Conclusion

In this paper we presented an algorithm that automatically compares semantic business process models with the aim of finding more general concepts. The algorithm works on business process models that are produced by a machine reading algorithm presented in our previous work. All models are uniform in their structure via their conformance with a 4D foundational ontology (BORO). We see the algorithm presented here as a necessary step toward mining general patterns from a large number of existing models developed across various domains – using such patterns to aid the design of new business processes.
Our approach is to compare business process data stored in the graph database repository and use a semantic similarity measure to determine whether two or more existing types in the ontological model can be subsumed by a more general type. In other words the underlying algorithm aims to calculate the Least Common Subsumer. The extracted hypertree is connected with the concepts that were used to calculate semantic similarity and stored into repository for future reference. The evaluation of the algorithm showed that the repository was enriched with a number of general concepts that are linked to the more specific concepts.

From a theoretical perspective this research contributes by bringing together the rich semantics of a foundational ontology with linguistic methods in order to generalise elements of business processes that are automatically discovered from semi-structured data.

From a practical perspective our work helps business process managers and engineers to semantically model their existing processes and to start identifying process patterns that can be reused or adapted in future BPM projects.

9. References