Labelling Business Entities in a Canonical Data Model

Nathali Ortiz Suarez
SAP SE
Dietmar-Hopp-Allee 16
69190 Walldorf
nathali.ortiz.suarez@sap.com

Jens Lemcke
SAP SE
Vincenz-Prienitz-Str. 1
76131 Karlsruhe
jens.lemcke@sap.com

Ulrike Pado
HFT Stuttgart
Schellingstr. 24
70174 Stuttgart
ulrike.pado@hft-stuttgart.de

Abstract

Enterprises express the concepts of their electronic business-to-business (B2B) communication in individual ontology-like schemas. Collaborations require merging schemas' common concepts into Business Entities (BEs) in a Canonical Data Model (CDM). Although consistent, automatic schema merging is state of the art, the task of labeling the BEs with descriptive, yet short and unique names, remains. Our approach first derives a heuristically ranked list of candidate labels for each BE locally from the names and descriptions of the underlying concepts. Second, we use constraint satisfaction to assign a semantically unique name to each BE that optimally distinguishes it from the other BEs.

Our system's labels outperform previous work in their description of BE content and in their discrimination between similar BEs. In a task-based evaluation, business experts estimate that our approach can save about 12% of B2B integration effort compared to previous work and about 49% in total.

1 Introduction

Businesses often exchange electronic messages like Purchase Orders, which contain compatible concepts (e.g., shipment dates and delivery address) that are however arranged and named differently in each company's ontology-like messaging standards (schemas). For instance, the two exemplary schemas shown on the left-hand side of Fig. 1 both speak about the delivery date, but use different phrases – "Current Scheduled Delivery" (node 10) and "Delivery Date/Time, estimated" (node 16). Misinterpretation is likely and may lead to delays and other financial losses.

The solution is to align the participating enterprises' schemas and find new, unique and appropriate (natural-language) names for the contained concepts, for all participants to use. A solution for the alignment task has been proposed in Lemcke et al. (2012): They create a CDM made up of BEs which can be visualised as clusters of equivalent nodes of the original schemas as visualized on the right-hand side of Fig. 1. This is similar to Ontology Merging (Shvaiko and Euzenat, 2013) except that the relation between the nodes is "part-of" and has to be maintained consistently.

As described in Lemcke et al. (2012), the only reliable source for correspondences between the schema nodes are the mappings business experts create when integrating two systems. Analysing the mappings shows that, for example, the delivery date is expressed in schema 1 by the value of node 8 in the "Date time" structure, together with the "Current scheduled delivery" qualifier (node 10). In schema 2, this corresponds to the combination of nodes 16 and 17. Therefore, BE I containing nodes 8, 10, 16 and 17 is created.

In this paper, we tackle the problem of automatically finding short, descriptive and unique natural-language labels for each of the BEs to replace the symbolic names F or I, based on the names and descriptions provided for each of the original nodes (see Table 1). The desired result are labels like

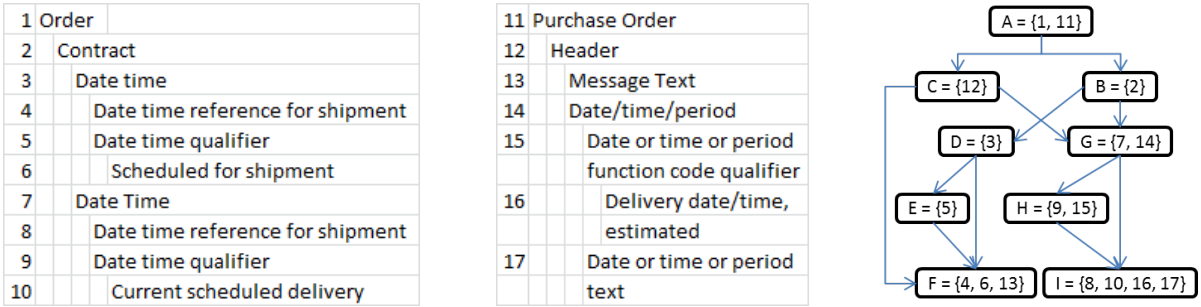

Figure 1: Two exemplary input schemas and corresponding Canonical Data Model (CDM)

BE	Node	Name of node	Description of node
F	4	Date time reference for shipment	To specify pertinent dates and times.
	6	Scheduled for shipment	
	13	Message Text	To provide a free-form format that allows the transmission of text information
I	8	Date time reference for shipment	To specify pertinent dates and times.
	10	Current scheduled delivery	
	16	Delivery date/time, estimated	Date and/or time when the shipper of the goods expects delivery will take place.
	17	Date or time or period text	To specify date, and/or time, or period.

Table 1: Exemplary BEs and nodes. Texts taken from B2B standards UN/EDIFACT (`http://www.unece.org/cefact/edifact/welcome.html`) and ASC X.12 (`http://www.x12.org/`).

Shipment Date and *Delivery Date*.

The labelling task is complicated by the limited vocabulary of the description data, since controlled terms from a strictly defined domain are used. For example, both BE description sets in Table 1 contain the words *date*, *shipment* or *scheduled*. Since we see fewer distinct content words than BEs, labels must be phrases. Also, we have to balance the need for short labels with specificity and discrimination amongst semantically similar BEs.

Further, reusing the same node defined by some schema template in different contexts is very common in B2B integration. For example, the date and time structures of node 4 and 8 in Table 1 can be interpreted either as a shipment or a delivery date, depending on whether they appear in conjunction with the qualifier node 6 (in BE F) or 10 (in BE I). This means that words and concepts introduced by different usage contexts of nodes are commonly used in BE descriptions.

Also, free text nodes like node 13 are commonly misused to store e.g. the shipment date. Both factors result in noise in the form of misleading words in the accumulated descriptions of a BE.

We clarify our assumptions about what defines a good label in Section 2. Based on these rules, our approach for labelling the CDM is described in Section 3. Note that the approach is completely domain- and mostly task-agnostic and could be used in other settings where short texts are involved. We present evaluation results with respect to label quality and time saved in Section 4.

2 Desiderata for Labels

An optimal labelling is reached when the following assumptions are true: Labels are natural language words or phrases that are:

Descriptive The label should state the concept of the BEs. Therefore, the concepts which are most frequently present in the names and descriptions of a BE are good label candidates.

Discriminative The label should state the distinguishing property of the BE. Therefore, the best candidates for labelling a BE are concepts which are frequently present in its names and descriptions, but not in the overall CDM.

Short The label should balance shortness (by Occam's razor) and specificity (to achieve uniqueness and discriminate between BEs).

Semantically Unique Two BEs must have non-synonymous labels. As the CDM has reference character for business experts, it is necessary to assign unique labels for unique BEs.

3 Labelling Business Entities

We use the approach developed in-house by (Dietrich et al., 2010). They introduce the tool pipeline shown in Fig. 2 to solve the labelling problem for the CDM. The Dietrich et al. approach generates label candidates from the node names and descriptions for each BE and validates them against a domain lexicon and search results in three search engines. However, due to data sparseness in both types of resources, correct label candidates are often erroneously rejected. Further, the approach conflates different senses of the same word. We address both of these issues below.

We also use a new strategy for labelling the CDM: First, we generate plausible label candidates for each BE and rank them heuristically. Second, we optimize globally, picking the set of labels for the CDM with the best overall ranks. This is similar to the global inference strategy, which recently has become increasingly popular (cf. work starting with Roth and Yih (2004)).

We now describe how we use and extend the tools from Fig. 2 to create labels with the properties defined in Section 2. Note that for both BE names and (possibly noisy) descriptions, processing is the same. We do, however, give more weight to the candidates extracted from the (cleaner) BE names. From here on, we use d_x as a placeholder to refer interchangeably to the names or the descriptions of the specific BE be_x.

Descriptive labels For descriptive labels, we need to find the most representative concept in a BE be_x. One strategy could be to look for domain terms which can be assumed to be relevant, but the Dietrich et al. results indicate that existing resources are too sparse for this. Therefore, we consider every term in the BE names and descriptions. To be agnostic of synonyms, our adapted synonym finder first extracts all possible meanings of each term t by retrieving the synsets $S_t = \{s_1, s_2, \ldots, s_n\}$ from WordNet (Fellbaum, 1998). Further, S_t is extended by the synsets of derivationally related forms of t as returned by WordNet. To increases the possibility of overlaps of the synsets of different, related terms, especially when used as different POS. The frequency of the synset s among the synsets of all terms of the names and descriptions d_x of the BE be_x, denoted as $f(s, d_x)$, indicates the relevance of s for describing be_x. We normalize the frequency over all be_x's synsets $S_{d_x} = \bigcup_{t \in d_x} S_t$ as in the term frequency (TF) approach by

$$tf(s, d_x) = \frac{f(s, d_x)}{\max\{f(s, d_x) : s \in S_{d_x}\}}.$$

In contrast to solely TF, the full TF/IDF approach did not yield satisfactory results: We found that since a BE's core concept may frequently appear in other BEs' descriptions due to re-use of nodes in different contexts and the misuse of free-text nodes, the IDF term was commonly very small and erroneously filtered out the true core concept.

For the final creation of labels, we express a synset s by the most frequent term t from d_x with $s \in S_t$ to adapt to the common technical terms of the domain.

Discriminative labels As there are fewer interesting words than BEs, word selection by TF does not produce unique labels, and phrases are needed. We use the description *The field represents the contract date representing the current scheduled delivery* to demonstrate how these are generated. First, nouns, verbs, adjectives and adverbs are identified as interesting words to build phrases. As an alternative design decision, each interesting term is then represented by its most frequent WordNet synset as described before and illustrated in Table 2. However, another alternative could be for example representing each interesting term by its first common hyperonym.

Second, our adapted phrase generator passes a sliding window over the text and considers all synset sequences in the window as possible candidates. With this window which was chosen heuristically, we both ensure some local coherence between the candidates and limit the numbers of possible combinations. For our running example we use a sliding window of size 4. We compute the relative distance of the synsets based on their position in the sentence. (E.g., *delivery* at position 11 and *current* at position 9 are two units apart.)

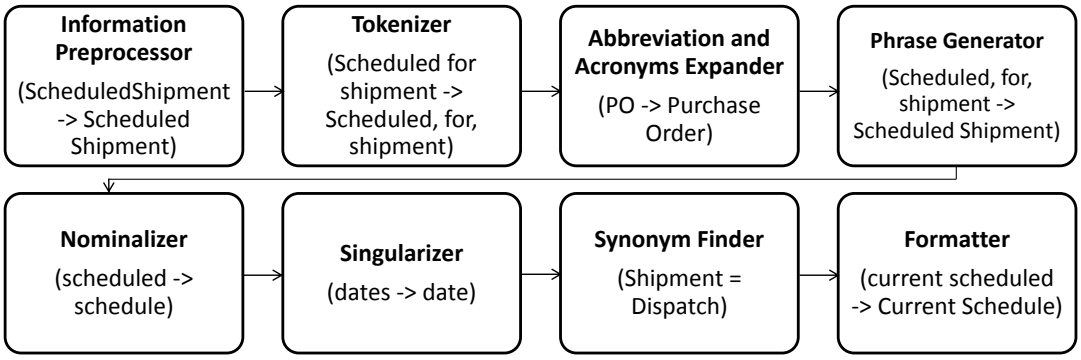

Figure 2: Tool pipeline for generating BE label candidates

Token	field	represents	contract	date	representing	current	scheduled	delivery
Position	2	3	5	6	7	9	10	11
POS	N	V	N	N	V	A	A	N
Synset	S1	S2	S3	S4	S2	S5	S6	S7

Table 2: Representation of each term as position in sentence, POS, and most frequent synset

Tag Pattern	**Example**
AN	Scheduled Delivery
NN	Reference Shipment
AAN	Current Scheduled Delivery
ANN	Added Tax Delivery
NAN	Reference Scheduled Delivery
NNN	Date Time Shipment
NPN	Reference for Shipment

Table 3: Justeson and Katz (1995) phrase patterns

The lower the relative distance, the more likely the phrase is to be useful, because it is present (almost) verbatim in the input. Third, for avoiding redundancy, we filter out synset sequences that contain duplicate synsets. Fourth, our adapted formatter chooses the most relevant word from the input sequences for each combination of synset and POS tag. The resulting phrase has to correspond to the POS tag sequences proposed in Justeson and Katz (1995) shown in Table 3. The input sequence in Table 2 yields phrases like *field representation*, *contract date representation*, *scheduled delivery*, *current scheduled delivery* and *current scheduled*. *Current scheduled* matches no pattern in Table 3, so it is changed to *current schedule*.

We estimate the quality of the phrases heuristically instead of checking against lexical resources. We use the length $le = |\mathbf{p}|$ of the phrase $\mathbf{p}$ to rank more specific phrases higher. We also consider the average frequency

$$\overline{wf} = \frac{\sum_{t \in \mathbf{p}} tf(t, d_x)}{le}$$

of the words of the phrase $\mathbf{p}$ in the names or descriptions of the BE be_x, favouring labels with more descriptive terms.

Short labels The previous step prefers relevant, but longer labels. We balance this preference with two measures that discourage long phrases: We consider the reciprocal of the average distance $\overline{di}^T = \frac{le-1}{di}$ of the words in a phrase, where di is the distance between the first and the last word of the compound in the original text, favouring short phrases taken literally from the text. The frequency $pf = tf(\mathbf{p}, d_x)$ of the phrase in the names or descriptions of be_x has a similar effect because longer phrases tend to be less frequent.

The final ranking of label candidates uses all four measures (length, average word frequency, inverse average word distance, and phrase frequency), each normalized over all candidates.

All weights are equal, except that we weight the measures for extracting phrases from the BE names twice as high as the measures for extracting from BE descriptions, since the names are defined by experts and contain less noise than was observed in the BE descriptions. This decision was supported by our analysis of results on the development set.

Semantically unique labels Finally, one of the locally generated phrases needs to be assigned to each BE, but not two BEs can get synonym labels. To solve this problem in a globally optimal way, we formulate the constraints and variables of a Constraint Satisfaction Problem (CSP). The CSP is solved by Choco 2.1.3 (choco Team, 2010), a very general constraint satisfaction framework.

Each BE be_x is represented by the variables $label$ (candidate phrases), $synsets$ (synset sequence for each phrase) and $rank$ (rank in terms of our heuristics).

A set of feasible tuples constraints ensures that $label$, $synsets$ and $rank$ are internally consistent for each BE be_x. Another two sets of all-different constraints ensure uniqueness among the values assigned to the $label$ and respectively to the $synsets$ variables, i.e., labels have to be unique both in terms of tokens and of concept. The system maximizes the formula $\sum_x rank_x$.

The complexity of the CSP depends most strongly on the size of the CDM, i.e., number b of BEs, and the window size w when generating phrases. The number of phrases, which make up the domains of the $label$ variables, depends exponentially on the window size and linearly on the length of names and descriptions. The CSP itself has exponentially many solutions depending on the number of BEs. So, the total worst-case complexity is $\mathcal{O}(2^{wb})$. In our case, with a $w = 5$ and $b = 25$ the computational time is approximately 3 hours and with the same w but $b = 38$ it is approximately 6.45 hours.

4 Evaluation

For evaluation, we compare to the baseline approach by Dietrich et al. (2010). We use 38 BEs that were unseen during the development of the tool pipeline. This data has the disadvantage of being proprietary, but there is not, to our knowledge, a comparable freely-available data set.

Our first objective is to establish the need for enforcing **unique labels**. Recall that our approach is designed to never assign the same label to different BEs. We automatically analysed the names proposed by the baseline approach, which assigns non-unique labels to 21% of the BEs. This is not acceptable in practice, since the point of the CDM is to allow unambiguous communication.

	Our	**BL**
Correct	70.3%	60.2%
Incorrect	29.7%	39.8%

Table 4: Descriptive and discriminative labels: Percentage of correct label-description pairings for our and the Baseline (BL) approach

The second part of our evaluation focuses on the **descriptive** and **discriminative** properties of our labels. This evaluation was done by ten novice users (due to the limited availability of experts). They assessed whether the label assigned to a BE correctly reflects its distinguishing features. In the survey, the participants answered 20 questions (ten for each approach). The participants saw the top-ranked BE label as generated by one of the systems, as well as the description of the input BE and the descriptions of semantically similar distractor BEs. If the participant chose the input description as best matching the label, we took that to mean that the label correctly distinguishes the semantics of the BE from the others.

The results of this survey are shown in Table 4. For the baseline approach, the participants chose the correct description for the label in 60.2% of the time, as opposed to 70.3% of the time for our approach. A X^2-test with a null hypothesis of chance assignment of correct and incorrect labels is significant at the 0.05 level; we conclude that our labels are more discriminative among BEs and describe BE content better than the labels returned by the baseline approach.

Finally, we present a task-based evaluation that was carried out with the help of B2B experts. Our objective here is to show that our system is useful in a real-world setting to the very group of people who are its intended users. Nine B2B experts estimated how much time and effort they would have saved creating the labels with the help of the output data of the approaches. The survey used five BEs and had three kinds of questions:

First, the participants were asked to create a label for a BE by hand, based on the names and descriptions available for it. These names and descriptions were also input to the systems.

Second, based on their manually created label, the participants estimated how much effort they could have saved in step 1 if they had had available

Effort Saved	Our	BL
$\geq 90\%$	8	4
90-75%	5	2
75-50%	5	5
50-30%	13	12
$\leq 30\%$	14	22
Avg (%)	49.2	37.1

Rank	Our	BL
1	36	9
2	26	19
3	21	24
4	18	27
5	19	26
6	15	31
Avg	3.02	3.99

Table 5: Task-based evaluation of label usefulness to experts: Result for evaluating effort saved (left) and label rank according to usefulness (right) of our and the baseline (BL) approach

the label candidates by one of the approaches. The participants chose one of five levels: more than 90% (when the label in step 1 is almost equal to the proposed candidates), between 90 and 75%, between 75 and 50%, between 50 and 30% and less than 30% (when the label is completely different).

Third, six model labels, three from each approach, had to be ranked in order of their usefulness for creating their label.

Table 5 shows the result for the effort-saved estimation on the left-hand side. We computed the average amount of effort saved by using the mid-point for each of the categories, e.g. 82.5 for the 90-75% category. Our approach saved 12.1 percentage points more expert effort than the baseline, and 49.2% of total effort. This corresponds to about four working hours (out of an eight-hour day). The baseline approach would allow the experts to save about three working hours, so using our approach saves an additional hour of (highly-qualified and highly-compensated) expert times.

The right-hand side of Table 5 shows the summarized results from the ranking task. Numerically, the experts ranked our proposals on average one rank higher than the baseline proposals. X^2-tests with the null hypothesis of an equal number of total observations in each rank found that the numerical differences for rank 1 and 6 are statistically significant at the 0.05 level. Overall, these results again illustrate that proposals given by our approach will be more useful for the experts in label creation than the baseline system.

5 Related Work

This paper is concerned with labelling a merged ontology in an unsupervised way given the node names and descriptions from the source ontologies. To our knowledge, this task is not commonly treated in the ontology merging literature.

In computational linguistics, our task is most comparable to the problem of assigning keywords or index terms that best describe a document's content (see, e.g., Kim et al. (2010)). However, our data is shorter, more repetitive and more ambiguous than running text from scientific publications or newspapers, and we have to obey the additional constraint of finding unique labels.

The labelling task is also somewhat reminiscent of the task of finding appropriate names for FrameNet framesets in the SemFinder system (Green and Dorr, 2004). Green and Dorr use WordNet synsets and glosses as their input data and rely heavily on WordNet's tree structure. This strategy is however infeasible for highly domain-specific texts like ours.

6 Conclusions

This paper proposed a method for labelling the BEs of a CDM by analysing the aggregated names and descriptions underlying the BEs, assuming that appropriate labels should be descriptive, discriminative, short and semantically unique.

Our strategy is very general and can be applied to other tasks inside and outside the ontology labelling domain. Several properties of the B2B domain challenged our implementation: Re-use and misuse of structural elements caused notable noise in the input data and the limited vocabulary of controlled terms means that the same relevant terms and concepts appear in multiple BEs.

We therefore applied the strategy of generating phrases as label candidates locally and then picking globally optimal label candidates. This strategy ensures unique labels, which are a core requirement in our domain.

Our evaluation showed that our labels are more descriptive of BE content and discriminate better among similar BEs than the baseline. A task-based evaluation with B2B experts, who are the intended users of the system, suggests potential effort savings in this crucial task of B2B integration of almost 50%, corresponding to four working hours out of an eight-hour work day.

References

choco Team. 2010. choco: an Open Source Java Constraint Programming Library. Research report 10-02-INFO, École des Mines de Nantes.

Michael Dietrich, Dirk Weissmann, Jörg Rech, and Gunther Stuhec. 2010. Multilingual extraction and mapping of dictionary entry names in business schema integration. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, pages 863–866.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Rebecca Green and Bonnie J Dorr. 2004. Inducing a semantic frame lexicon from wordnet data. In *Proceedings of the 2nd Workshop on Text Meaning and Interpretation*, pages 65 – 72.

John S. Justeson and Slava M. Katz. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26.

Jens Lemcke, Gunther Stuhec, and Michael Dietrich. 2012. Computing a canonical hierarchical schema. In *Proceedings of the I-ESA Conferences Volume 5*, pages 305–315.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Conference on Natural Language Learning*, pages 1–8.

Pavel Shvaiko and Jérôme Euzenat. 2013. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176.