Ontological Blending in DOL

Oliver Kutz, Till Mossakowski, Joana Hois, Mehul Bhatt, John Bateman [1]

Abstract. We introduce ontological blending as a method for combining ontologies. Compared with existing combination techniques that aim at integrating or assimilating categories and relations of thematically related ontologies, *blending* aims at creatively generating (new) categories and ontological definitions; this is done on the basis of input ontologies whose domains are thematically distinct but whose specifications share structural or logical properties. As a result, ontological blending can generate new ontologies and concepts and it allows a more flexible technique for ontology combination compared to existing methods.

Our approach to computational creativity in conceptual blending is inspired by methods rooted in cognitive science (e.g., analogical reasoning), ontological engineering, and algebraic specification. Specifically, we introduce the basic formal definitions for ontological blending, and show how the distributed ontology language DOL (currently being standardised within the OntoIOp—Ontology Integration and Interoperability—activity of ISO/TC 37/SC 3) can be used to declaratively specify blending diagrams.

1 Introduction

Well-known techniques directed towards unifying the semantic content of different ontologies, namely techniques based on matching, aligning, or connecting ontologies, are ill-suited to either re-use (proven) axioms from one ontology in another or generate new conceptual schemas from existing ontologies, as it is suggested by the general methodology of conceptual blending introduced by Fauconnier and Turner [11]: here, the blending of two thematically rather different *conceptual spaces* yields a new conceptual space with emergent structure, selectively combining parts of the given spaces whilst respecting common structural properties.[2] The 'imaginative' aspect of blending is summarised as follows [39]:

> [...] the two inputs have different (and often clashing) organising frames, and the blend has an organising frame that receives projections from each of those organising frames. The blend also has emergent structure on its own that cannot be found in any of the inputs. Sharp differences between the organising frames of the inputs offer the possibility of rich clashes. Far from blocking the construction of the network, such clashes offer challenges to the imagination. The resulting blends can turn out to be highly imaginative.

A classic example for this is the blending of the concepts *house* and *boat*, yielding as most straightforward blends the concepts of a *houseboat* and a *boathouse*, but also an *amphibious vehicle* [16].

In the almost unlimited space of possibilities for combining existing ontologies to create new ontologies with emergent structure, conceptual blending can be built on to provide a structural and logic-based approach to 'creative' ontological engineering. This endeavour primarily raises the following two challenges: (1) when combining the terminologies of two ontologies, the shared semantic structure is of particular importance to steer possible combinations. This shared semantic structure leads to the notion of base ontology, which is closely related to the notion of 'tertium comparationis' found in the classic rhetoric and poetic theories, but also in more recent cognitive theories of metaphor (see, e.g., [23]); (2) having established a shared semantic structure, there is typically still a huge number of possibilities that can capitalise on this information in the combination process: here, optimality principles for selecting useful and interesting blends take on a central position.

We believe that the principles governing ontological blending are quite distinct from the rather informal principles employed in blending phenomena in language or poetry, or the rather strict principles ruling blending in mathematics, in particular in the way formal inconsistencies are dealt with. For instance, whilst blending in poetry might be particularly inventive or imaginative when the structure of the basic categories found in the input spaces is almost completely ignored, and whilst the opposite, i.e., rather strict adherence to sort structure, is important in areas such as mathematics in order to generate meaningful blends[3], ontological blending is situated somewhere in the middle: re-arrangement and new combination of basic categories can be rather interesting, but has to be finely controlled through corresponding interfaces, often regulated by or related to choices found in foundational or upper ontologies.

We start with a discussion of alignment, matching, analogical reasoning, and conceptual blending, vis-à-vis ontological blending. The core contributions of the paper[4] can be summarised as follows; we:

- give an abstract definition of ontological blendoids capturing the basic intuitions of conceptual blending in the ontological setting;
- provide a structured approach to ontology languages, in particular to OWL-DL[5], by employing the OWL fragment of the distributed ontology language DOL for blending, namely DOL-OWL. This combines the simplicity and good tool support for OWL with the more complex blending facilities of OBJ3 [17] or Haskell [25];
- analyse the computational and representational issues that blending with ontology languages raises, and outline some of the first optimality principles for ontological blending;

[1] Research Center on Spatial Cognition (SFB/TR 8), University of Bremen, Germany. Corresponding author: okutz@informatik.uni-bremen.de

[2] The usage of the term 'conceptual space' in blending theory is not to be confused with the usage established by Gärdenfors [13].

[3] For instance when creating the theory of transfinite cardinals by blending the perfective aspect of counting up to any fixed finite number with the imperfective aspect of 'endless counting' [34].

[4] This paper elaborates on ideas first introduced in [20].

[5] In the remainder of this paper we refer to OWL-DL Version 2 by just OWL. See http://www.w3.org/TR/owl2-overview/

The contributions are illustrated in detail with a fully formalised example of an ontological blend, involving signs (signposts) and forests.

2 Ontology Alignment and Conceptual Blending

For a given domain, often several ontologies exist which need to be related in order to achieve coverage of the required knowledge. For instance, heterogeneous sources may provide ontological information on the same kind of data, and their information needs to be integrated with each other. Various kinds of relations between these types of ontologies have been studied in the literature, amongst them mapping and matching, alignment, coordination, transformation, translation, merging, reconciliation, and negotiation (cf. [6]). Some of these techniques, in particular matching and alignment, are typically based on statistical approaches and similarity measures [24, 10].[6]

From these techniques, alignments are most closely related to our present purpose because they can be seen as a strict, i.e., 'uncreative', version of blending. Alignments completely identify or separate information, in particular, they try to find semantically related concepts or relations from two given ontologies. They seek out commonalities between these concepts or relations by inspecting surface data, e.g., concept and relation names. However, they typically ignore their logical information, namely the axiomatisations of the ontologies. The quality of detected alignments is typically assessed by comparison to a previously defined gold-standard based on standard precision and recall methods.[7] In general, alignments are most useful for combining ontologies that specify thematically closely related domains.

The alignment operation between two ontologies was first formalised from a category-theoretic standpoint in [41], using pushouts and colimits, and further refined in [26]. A pushout links two given ontologies using a common interface theory. While the ontologies are disjointly united, the two copies of the common interface theory are identified. For example, if ontology O_1 features a concept Human, while O_2 provides Person, a corresponding concept should occur in the common interface theory and be mapped to Human and Person, respectively. The effect is that in the alignment (formalised as a pushout), Human and Person are identified. In contrast, if concepts do not appear in the common interface, they are kept apart, *even if they happen to have the same name* (cf. Bank in the example).

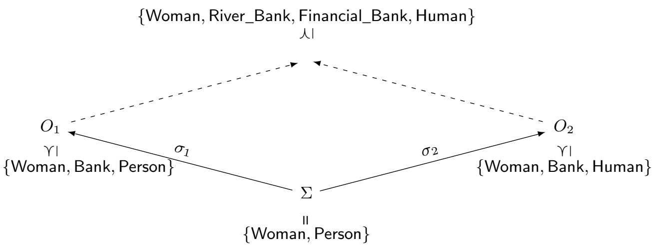


Figure 1. V-alignment: integration through interface

This construction, called V-alignments, can deal with basic alignment problems such as **synonyms** (identifying different symbols with the same meaning) and **homonyms** (separating (accidentally) identical symbols with different meaning)—see Fig. 1. Alignments, however, can support only these basic types of relations between two ontologies having thematically overlapping domains. Combinations of thematically different ontologies can easily become more complex, for instance, when dealing with **analogies** (relating different symbols based on their similar axiomatisation), **metaphors** (blending symbols from one domain into another and impose the axiomatisation of the first on the second), **pataphors** (blending and extending two domains with each other), or **conceptual blending** (blending and combining two domains for the creation of new domains). In contrast to alignments, blending thus combines two potentially thematically unrelated ontologies in a way such that new structure can emerge. Below, we define and formalise this blending operation accordingly.

In [35], conceptual blending is implemented in terms of analogy finding applied to an automatic text generation system. Particularly, for metaphorical phrasing, the tool *jMapper* compares the instances of two given input domains with each other and calculates the similarity between instances of the source and the target domain. This is based on shared properties and relationships of the domain's instances, for which thresholds can be varied. However, the *jMapper* tool does not aim at creating 'new' domains. It only works with instance definitions as input domains in a proprietary format rather than re-using standardised ontology languages.

In [25], blending is based on structural aspects of two different domains. The example of blending boat and house is here based on image schemata, namely, categories and relations from the house and boat domains are related to particular image schemata such as *container* and *surface*. The image schemata are used as an abstraction necessary for blending two domains. The boat and house example is implemented using Haskell type classes, which, however, results in rigidly blended classes for houseboat and boathouse. For instance, only a 'boat' can be an 'inhabitant' of a 'boathouse'. Any other (conceptually possible) type, such as a caretaker residing in a boathouse, contradicts this definition. Conceptual blending in general does not exhibit this kind of strong restriction.

In [16], conceptual blending is formalised categorically, focusing on the structural aspects of the blending process. In the following, we adapt this approach to ontological engineering.

3 Introducing Ontological Blending

Goguen has created the field of *algebraic semiotics* which logically formalises the structural aspects of semiotic signs, sign systems, and their mappings [15]. In his joint work with Fox Harrell [16], algebraic semiotics has been applied to user interface design and blending. Algebraic semiotics does not claim to provide a comprehensive formal theory of blending—indeed, Goguen and Harrell admit that many aspects of blending, in particular concerning the meaning of the involved notions, as well as the optimality principles for blending, cannot be captured formally. However, the structural aspects *can* be formalised and provide insights into the space of possible blends.

Goguen defines semiotic systems to be algebraic theories that can be formulated by using the algebraic specification language OBJ [17]. Moreover, a special case of a semiotic system is a *conceptual space*: it consists only of constants and relations, one sort, and axioms that define that certain relations hold on certain instances.

As we focus on standard ontology languages, namely OWL and first-order logic, we here replace the logical language OBJ. As structural aspects in the ontology language are necessary for blending, we augment these languages with structuring mechanisms known from

[6] Ontology matching and alignment based on such methods is an established field on its own having yearly competitions since 2004 (see `http://oaei.ontologymatching.org/`).

[7] See [19] for an extensive analysis. The lack of semantics involved in such an evaluation process has been clearly articulated already in [9].

algebraic specification theory [27]. This allows to translate most parts of Goguen's theory to these ontology languages. Goguen's main insight has been that semiotic systems and conceptual spaces can be related via *morphisms*, and that blending is comparable to *colimit* construction. In particular, the blending of two concepts is often a *pushout* (also called *blendoid* in this context). Some basic definitions:

An OWL **signature** consists of sets of class names, role names, and individual names. An OWL **signature morphism** between two OWL signatures consists of three mappings between the respective sets. OWL sentences over a given signature Σ are defined as in [22], e.g., subsumptions between classes, role hierarchies, and instances of classes and roles, etc. OWL models provide a domain of individuals and interpret classes as subsets, roles as binary relations, and individuals as elements of the domain. Satisfaction of sentences in a model is defined in a standard way, see [22] for details. Moreover, given a signature morphism $\sigma : \Sigma_1 \to \Sigma_2$ and a Σ_2-model M_2, the **reduct** $M_2|_\sigma$ is the Σ_1-model that interprets a symbol by first translating it along σ and then looking up the interpretation in M_2.

On top of this, we define the language DOL-OWL and its model-theoretic semantics as follows.[8] A DOL-OWL ontology O can be

- a **basic** OWL theory $\langle \Sigma, \Gamma \rangle$; Σ is a signature, Γ a set of Σ-sentences, with $\mathbf{Mod}(\langle \Sigma, \Gamma \rangle)$ containing all Σ-models satisfying Γ;
- a **translation**, written O **with** σ, (where $\sigma : \Sigma_1 \to \Sigma_2$) with $\mathbf{Mod}(O \textbf{ with } \sigma) = \{M \in \mathbf{Mod}(\Sigma_2) \mid M|_\sigma \in \mathbf{Mod}(O)\}$;
- a **union**, written O_1 **and** O_2, of ontologies over the same signature, with $\mathbf{Mod}(O_1 \textbf{ and } O_2) = \mathbf{Mod}(O_1) \cap \mathbf{Mod}(O_2)$[9];
- a **hiding**, written O **hide** σ, with $\mathbf{Mod}(O \textbf{ hide } \sigma) = \{M|_\sigma \mid M \in \mathbf{Mod}(O)\}$.

A DOL-OWL library statement can be

- an ontology definition **ontology** O_NAME = O; or
- a interpretation, written **interpretation** $INT\_NAME : O_1$ to $O_2 = \sigma$.

An interpretation is **correct**, if σ is a theory morphism from O_1 to O_2, that is, for every O_2-model M_2, its reduct $M_2|_\sigma$ is an O_1-model. This definition provides a structural approach in DOL-OWL, that can be compared with instantiation of type variables in Haskell and type casting in OBJ3.

Since in some blends, not the whole theory can be mapped, Goguen [15] introduces partial signature morphisms. Here, we follow a common idea in category theory and model partial theory morphisms $\sigma : T_1 \multimap\!\!\to T_2$ as spans

$$T_1 \xleftarrow{\;\sigma_-\;} dom\ \sigma \xrightarrow{\;\sigma_+\;} T_2$$

of ordinary (total) theory morphisms satisfying a well-definedness condition; this has the advantage of keeping the theory simple. σ_- is the inclusion of $dom\ \sigma$ (the domain of σ) into T_1, while σ_+ is the action of the partial theory morphism. If σ_- is an isomorphism, we say that σ is total, it can then be identified with the ordinary morphism $\sigma_+ \circ \sigma_-^{-1} : T_1 \to T_2$:

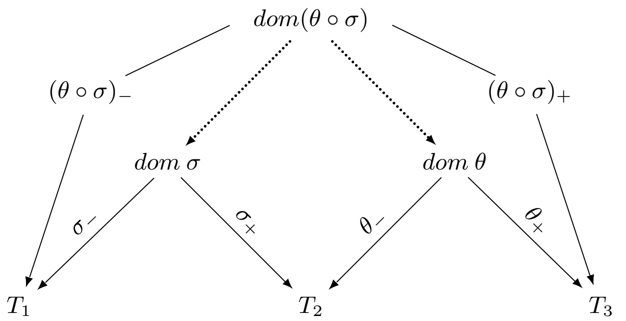
$$T_1 \xrightarrow{\;\sigma_-^{-1}\;} dom\ \sigma \xrightarrow{\;\sigma_+\;} T_2$$

8 The definition of DOL-OWL as given here corresponds essentially to the fragment of the distributed ontology language DOL that homogeneously uses OWL modules. The full DOL language however comprises several additional features, and supports a large number of ontology languages, see [32] for a presentation of the full semantics.

9 Unions over different signatures can be modelled using translations.

The well-definedness condition for partial theory morphisms $\sigma : T_1 \multimap\!\!\to T_2$ is similar to but more general than that for ordinary theory morphisms: for each T_2-model M_2, its reduct $M_2|_{\sigma_+}$ must be "somehow" a T_1-model. The "somehow" can be made precise as follows: for each T_2-model M_2, there must be a T_1-model M_1 such that $M_1|_{\sigma_-} = M_2|_{\sigma_+}$. Equivalently, $\sigma_+ : (T_1 \textbf{ hide } \sigma_-) \to T_2$ is an ordinary theory morphism (note that the models of T_1 **hide** σ_- are precisely those models that are σ_--reduct of some T_1-model).

We now recall some notions from category theory, see [1, 41] for further details. A **diagram** D consists of a graph of ontologies $(D_i)_{i \in |D|}$ and total theory morphisms $(D_m : D_i \to D_j)_{m \in D}$ among them. Partial theory morphisms can easily be dealt with: diagrams just get a little larger when spans are used. For a diagram D, a **partial sink** consists of an ontology O and a family of partial theory morphisms $(\mu_i : D_i \multimap\!\!\to O)_{i \in |D|}$. A **sink** is a partial sink consisting of total morphisms only. A partial sink is an epi-sink, if $f \circ (\mu_i)_- = g \circ (\mu_i)_-$ for all $i \in |D|$ implies $f = g$. A partial sink is **weakly commutative** if all emerging triangles commute weakly, i.e., for all $m : i \to j \in D$, we have that $D_m \circ \mu_i = \mu_j$ as partial morphisms. Such compositions of partial morphisms are obtained by pullback:



For total sinks, weak commutativity amounts to ordinary commutativity; the sink in this case is called a **co-cone**. A co-cone is a **colimit**, if it can be uniquely naturally embedded into any co-cone (hence, it can be seen as a minimal co-cone). [1] also show that colimits are epi-sinks.

We now give a general definition of ontological blending capturing the basic intuition that a blend of input ontologies shall partially preserve the structure imposed by base ontologies, but otherwise be an almost arbitrary extension or fragment of the disjoint union of the input ontologies with appropriately identified base space terms.

Definition 1 (Ontological Base Diagram) *An **ontological base diagram** is a diagram D for which the minimal nodes $(B_i)_{i \in D_{min} \subseteq |D|}$ are called **base ontologies**, the maximal nodes $(I_j)_{j \in D_{max} \subseteq |D|}$ called **input ontologies**, and where the partial theory morphisms $\mu_{ij} : B_i \multimap\!\!\to I_j$ are the **base morphisms**. If there are exactly two inputs I_1, I_2, and one base $\mathcal{B}$, the diagram D is called **classical** and has the shape of a V (for total morphisms) or W (for partial morphisms). In this case, $\mathcal{B}$ is also called the **tertium comparationis**.*

The basic, i.e., classical, case of an ontological base diagram with total morphisms is illustrated in the lower part of Fig. 2. In general, however, ontological blending can deal with more than one base and two input ontologies. [8], for instance, discusses the example of blending the input domains *politics*, *American culture*, and *sports*, in order to create the metaphor "He's a guy who was born on third base and thinks he hit a triple." [8, p. 172] (a criticism of George Bush).

Definition 2 (Ontological Blendoid) *Let D be a base diagram. A*

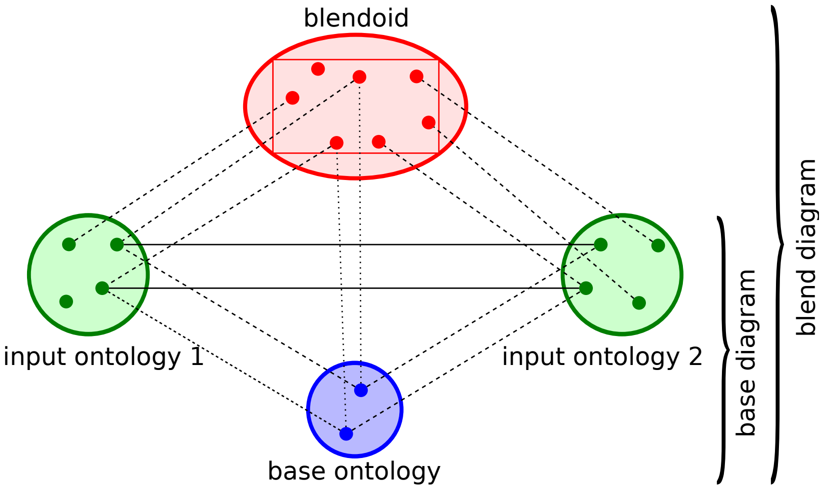


Figure 2. The basic integration network for blending: concepts in the base ontology are first refined to concepts in the input ontologies and then selectively blended into the blendoid.

blendoid $\mathfrak{B}$ *for* D *is a partial sink of* signature morphisms *over* D. *A blendoid is called*

- ***axiom-preserving**, if the signature morphisms of the partial sink are all theory morphisms;*
- ***closed**, if it is a (partial) epi-sink (which basically means that the blend is generated by the diagram), otherwise **open**;*
- ***total**, if the partial sink is a sink;*
- ***commutative**, if it is (weakly) commutative;*
- ***strict**, if it is a colimit (colimits are always epi-sinks, so closed).*

Here, axiom preservation, totality and commutativity can also hold to a certain degree. Consider the percentage of: signature morphisms that are theory morphisms (resp. total); and diagrams that commute.

Further note that an axiom-preserving strict blend where the base diagram has the form of a V and the base ontology is just a signature is nothing else but a V-alignment. Note that open blends might additionally import ontologies with new relevant signature.

Two crucial aspect of blends are (1) morphisms within the base diagram as well as into the blend diagram can be partial, and (2) the structure of the blend might partially violate the shared structure of the inputs ('violation of structure mapping').

In practice, open blends will typically be constructed by first generating a closed blend, and then subsequently aligning this with a new (thematically related) input ontology. In particular, this construction can be applied by aligning two different closed blends $\mathfrak{B}_1$ and $\mathfrak{B}_2$ obtained through the same base space $\mathcal{B}$ (here new signature elements can be created in the new colimit). For instance, we can align the blended ontologies for BoatHouse and HouseBoat by introducing houseboats as residents of boathouses. This **completion** by alignment or import can be seen as an analogue to the 'running of the blend' as it is discussed in conceptual blending [11].

Clearly, unless we construct a strict blendoid with a rather 'strong' base ontology, due to partiality there will always be exponentially many possibilities for the blend. Moreover, there are obviously infinitely many open blends regardless of partiality and the structure of the base. For instance, in the House and Boat blending formalised in [16], there are, in our terminology, 48 blendoids over a fixed base diagram that are axiom preserving, commutative and closed.[10]

[10] Note that this differs from the (slightly inconsistent) terminology in [16].

4 Computational and Representational Challenges

Conceptual blending has been proposed as a possible solution to get a handle on the notion of computational creativity [35]. The most sophisticated implementation to date related to blending probably is the tool described in [16] for the automated generation of poems. To create similar tools specifically dedicated to the realm of ontology, we have to address at least the following three issues:

1. The representational layer for blending needs to be specialised to ontology languages, in particular to one-sorted languages such as OWL, and languages such as Common Logic[11].
2. Given a couple (or a finite number of) ontologies, strategies are required to compute (rather than assume or handcraft) the common base ontology together with corresponding morphisms.
3. Given an ontological base diagram, techniques and heuristics are required that select interesting or useful blendoids according to genuine ontological principles. In particular, this requires new ranking and optimality principles.

We have addressed the first item already in the previous section: the language DOL-OWL allows for a structured specification of blend diagrams. Note that, more generally, mixed blend diagrams can be specified in the DOL language combining, besides several other ontology languages, first-order and OWL ontologies (see [28]). We next briefly discuss items 2. and 3.

4.1 Computing the Tertium Comparationis

To find candidates for base ontologies that could serve for the generation of ontological blendoids, much more shared semantic structure is required than the surface similarities that alignment approaches rely on. The common structural properties of the input ontologies that are encoded in the base ontology are typically of a more abstract nature. The standard example here relies on *image schemata*, such as the notion of a *container* mentioned earlier (see also [25]). Thus, in particular, foundational ontologies can support such selections. In analogical reasoning, 'structure' is (partially) mapped from a source domain to a target domain [12, 38]. Intuitively, then, the operation of

[11] See `http://common-logic.org/`

computing a base ontology can thus be seen as a bi-directional search for analogy.

We briefly discuss three promising candidates for this operation:

(1) **Ontology intersection:** [33] has studied the automatisation of theory interpretation search for formalised mathematics, implemented as part of the Heterogeneous Tool Set (HETS, see below). [29] applied these ideas to ontologies by using the ontologies' axiomatisations for finding their shared structure. Accidental naming of concept and role names is deliberately ignored and such names are treated as arbitrary symbols (i.e., any concept may be matched with any other). By computing mutual theory interpretations between the inputs, the method allows to compute a base ontology as an *intersection* of the input ontologies together with corresponding theory morphisms. While this approach can be efficiently applied to ontologies with non-trivial axiomatisations, lightweight ontologies are less applicable, e.g., 'intersecting' a smaller taxonomy with a larger one clearly results in a huge number of possible taxonomy matches [29]. In this case, the following techniques are more appropriate.

(2) **Structure-based ontology matching:** [37] address the problem that matching and alignment approaches are typically restricted to find simple correspondences between atomic entities of the ontology vocabulary. They define a number of *complex correspondence patterns* that can be used together with standard alignments in order to relate complex expressions between two input ontologies. For instance, the 'Class by Attribute Type Pattern' may be employed to claim the equivalence of the atomic concept $\mathsf{PositiveReviewedPaper}$ in ontology O_1 with the complex concept $\exists\mathsf{hasEvaluation.Positive}$ of O_2. Such an equivalence can be taken as an axiom of the base ontology; note, however, that it could typically not be found by intersecting the input ontologies. Giving such a library of design patterns may be seen as a variation of the idea of using image schemata.

(3) **Analogical Reasoning:** *Heuristic-driven theory projection* is a logic-based technique for analogical reasoning that can be employed for the task of computing a common generalisation of input theories. [38] establish an analogical relation between a source theory and a target theory (both first-order) by computing a common generalisation (called 'structural description'). They implement this by using anti-unification [36]. A typical example is to find a generalisation (base ontology) formalising the structural commonalities between the Rutherford atomic model and a model of the solar system. This process may be assisted by a background knowledge base (in the ontological setting, a related domain or foundational ontology). Indeed, this idea has been further developed in [30].

4.2 Selecting the Blendoids: Optimality Principles

Having a common base ontology (computed or given), there is typically a large number of possible blendoids. For example, even in the rather simple case of combining House and Boat, allowing for blendoids which only partially maintain structure (called *non-primary* blendoids in [16]), i.e., where any subset of the axioms may be propagated to the resulting blendoid, the number of possible blendoids is in the magnitude of 1000. Clearly, from an ontological viewpoint, the overwhelming majority of these candidates will be rather meaningless. A ranking therefore needs to be applied on the basis of specific ontological principles. In conceptual blending theory, a number of **optimality principles** are given in an informal and heuristic style [11]. While they provide useful guidelines for evaluating natural language blends, they do not suggest a direct algorithmic implementation, as also analysed in [16]. Moreover, the standard blending theory of [11] does not assign types, which might make sense in the case of linguistic blends where type information is often ignored. A typical example of a type mismatch in language is the operation of *personification*, e.g., turning a boat into an 'inhabitant' of the 'boathouse'. However, in the case of blending in mathematics or ontology, this loss of information is often rather unacceptable: to the opposite, a fine-grained control of type or sort information is of the utmost importance here.

Optimality principles for ontological blending will be of two kinds. (1) purely *structural/logical principles*: as introduced in Sec. 3, these will extend and refine the criteria as given in [16], namely **degree of commutativity** of the blend diagram, **type casting** (preservation of taxonomical structure), **degree of partiality** (of signature morphisms), and **degree of axiom preservation**. The relative ranking and importance of these metrics, however, will remain a case-by-case decision. In the context of OWL, typing needs to be replaced with preservation of specific axioms encoding the taxonomy. (2) *heuristic principles*: unlike the categorical modelling of alignments, blendings can often not be adequately described by a pushout operation. Some diagrams may not commute, and a more fine-grained control is required. This particularly explains why Goguen uses $3/2$ pushouts to specify blending [15]. Generalising blendoids to be $3/2$ pushouts allows for the integration of certain optimality principles in the blending process, namely an ordering of morphisms allowing to specify their quality (for instance in terms of their degree of partiality and type violation). Essentially, this introduces preference orders on possible morphisms, which can further be regulated by specific ontological principles. One candidate for regulating such preference orders, extending the purely structural optimality principles, would be adherence to the OntoClean methodology [18].

Existing Tool Support. For carrying out blending experiments using OWL, we use the DOL-OWL language and the Heterogeneous Tool Set HETS [31] which provides a prototypical implementation of the full DOL language.[12] DOL-OWL allows for writing OWL ontologies using Manchester syntax [21] (hence they can also be imported from common tools like Protégé), and DOL-OWL provides *interpretations* in the style of OBJ views that relate logical theories (here: OWL ontologies), using interpretations of theories. Interpretations are also used to build up the blending diagrams. Moreover, HETS can compute colimits of such diagrams, as well as approximations of co-limits in the case where the input ontologies live in different ontology languages [7]. These features are essential for the implementation of the example discussed next.

5 Example: Blending Forests and Signs

We briefly describe the theories of signs, forests, and their blends informally, followed by a sketch of the formal specifications of the involved ontologies and their blending.

5.1 An Informal Theory of Forests and Signs

Signs are defined as "(for information / warning) a piece of paper, wood or metal that has writing or a picture on it that gives you information, instructions, a warning, etc.: a road / traffic sign; a shop / pub sign" (taken from Oxford Advanced Learner's Dictionary). In the signage theory, signs are physical artefacts, which are defined by their colour, shape, and location, and they depict a small amount of symbols, i.e., the number of symbols on a sign may not exceed seven

[12] HETS is available under `www.dfki.de/cps/hets`. For more information on DOL and the ISO standardisation effort OntoIOp visit `http://ontolog.cim3.net/cgi-bin/wiki.pl?OntoIOp`

Figure 3. Examples for Sign (top-left), Forest (bottom-left), ForestSign (top-right), and SignForest (bottom-right) [taken from various sources]

items (which is an estimated amount of items). These symbols convey information, which may point to other objects. But also shape or colour can convey information. Signs can in principle be classified into different types of signs, such as road sign or warning sign. Forests are defined as "complex ecological systems in which trees are the dominant life form" (taken from Encyclopaedia Britannica). In the forest theory, forests are natural groups of 'soil, plant, and animal life' with a high density of trees. Here, forests have to contain at least 100 trees (which is an estimated count for simplicity). They can again be classified into subtypes, such as rainforest or tropical forest.

Blending the theories of signs and forests can result in diverse new theories. A blend *forest sign* can, for instance, describe (a) a sign pointing to a forest (by tree icons or the name of the forest), (b) a sign with the shape of a tree, or (c) a sign located in a forest. A blend *sign forest* can, for instance, (a) describe road sign clutter (a 'sign forest'), (b) describe a sign forest that consists of forest signs, or (c) identify the Sign Post Forest (see `http://www.signpostforest.com`). Fig. 3 shows examples of a sign and a forest together with the blends forest sign and 'sign forest' (road sign clutter).

Different blends are mostly based on different base ontologies. The base ontology can specify basic aspects on which the input ontologies for forests and signs agree. For instance, a base ontology can define a category (container) that consists of many entities of the same kind that are essential to determine the category's type. In detail, a sign consists of symbols that determine the sign's type while the forest consists of trees that determine the forest's type. Alternatively, a base ontology can specify that you can get lost in a certain environment. In detail, you can get physically lost in forests, i.e., you do not find your way out, and you can get mentally lost in signs, i.e., you do not see the information conveyed. Furthermore, a base ontology may specify constraints on both input ontologies, such as every forest has more trees than signs have symbols and, consequently, it is not allowed to blend forest to sign and tree to symbol in the same blendoid. Again, the base ontology specification may be guided by foundational ontologies, as described above.

5.2 Ontologies of Forest, Signage and SignForest in DOL-OWL

The two input ontologies in Fig. 4 show parts (modules) of the specifications of the Signage and Forest theory.[13] They formalise signs and forests as described in the previous section. Arrows indicate relationships between classes (i.e., the axiomatisation of the ontologies), thick lines indicate class mappings given by the theory morphisms between the base ontology, the input ontologies and the blend, light grey classes and relations are conservative extensions, which are relevant for the calculation of the colimit. The essential information in the base ontology that can lead to the signforest blendoid specifies a container class that contains objects that have a certain location. From here, partial theory morphisms are defined as interpretations in DOL-OWL that relate classes from the base ontology to classes from the input ontology (along the thick lines, cf. Fig. 4), resulting in the base diagram. Those parts of the base ontology that are not related to parts of the input ontologies are hidden by these partial theory morphisms. However, in order to calculate the colimit that creates the signforest blendoid, these hidden parts are revealed by conservatively extending the input ontologies and making the theory morphisms total, as indicated in Section 3. For example, the morphism from the base ontology to the forest ontology hides the relation hasLocation, which is not specified in the original forest ontology, but the relation then gets related to growsOn in the conservatively extended forest ontology.

Based on the interpretations from Signage and Forest, the input ontologies are blended into the blendoid SignForest by calculating the colimit of the two input ontologies resulting in a tame blendoid. In detail, Forest is identified as Forest in the blendoid. It contains the class Sign, which is mapped to Tree. The typecast of this mapping leads to a 'treeification' of signs, similar to the 'personification' of boats as inhabitants of boathouses. According to the base ontology, these 'treeified' signs have a location (hasLocation) at a certain abstract PhysicalSupport. Note that the blendoid specifies sign forests to contain at least 100 signs, whilst its conceptualisation allows a smaller amount, i.e., the resulting blendoid should be further refined.

6 Discussion and Future Work

Our work in this paper follows a research line in which blending processes are primarily controlled through mappings and their properties [14, 12, 40, 35]. By introducing blending techniques to ontology languages, we have provided a new method which allows to combine two thematically different ontologies in order to re-use axioms in other ontologies and to create a new ontology, the blendoid, describing a newly created domain. The blendoid creatively mixes information from both input ontologies on the basis of structural commonalities of the inputs and combines their axiomatisation.

Ontological blending can serve as an exploratory tool for semantic information retrieval systems (e.g., in medicine) [4]; here, ontological blending will provide the capability to automatically create blend-ontologies from multiple input ontologies that each reflect a certain domain of interest and expertise, e.g., doctors, pharmacists, nurses, each having a different *perspective* on treatment procedures and available information, but with certain shared conceptualisations. Similarly, blending serves to fulfill a creative function within design systems where multi-perspective semantics and reasoning about design concepts is essential [3].

[13] The DOL-OWL specifications is available at: `www.informatik.uni-bremen.de/~okutz/blending/blending.html`

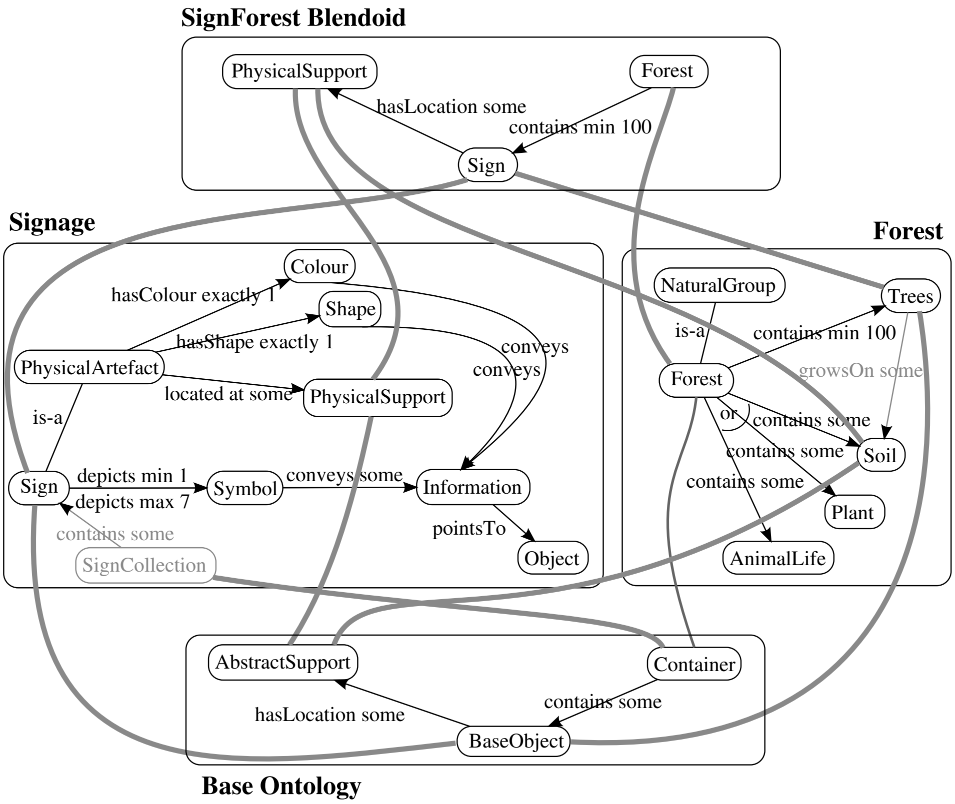


Figure 4. Blending Forest and Signage resulting in the SignForest blend

We have illustrated that the tool HETS and the DOL language [32] (here the DOL-OWL fragment discussed above) provide an excellent starting point for developing the algorithmic side of the theory further. They: (1) support various ontology language and their heterogeneous integration [27]; (2) allow to specify theory interpretations and other morphisms between ontologies [28]; (3) support the computation of colimits as well as the approximation of colimits in the heterogeneous case [7]; (4) provide (first) solutions for automatically computing a base ontology through ontology intersection [29].

However, to make ontological blending feasible in practice, all of these aspects need to be further refined, as discussed above. This concerns primarily the ontological optimality principles (e.g., for semantic completeness and related optimisation heuristics [5]) as well as means for computing common base ontologies [2]. Both issues are almost completely new research questions in ontology research, and we here gave a first analysis and partial answers to them.

Acknowledgements

Work on this paper has been supported by the DFG-funded collaborative research centre SFB/TR 8 'Spatial Cognition'.

REFERENCES

[1] J. Adámek, H. Herrlich, and G. Strecker, *Abstract and Concrete Categories*, Wiley, New York, 1990.
[2] Mehul Bhatt, Andrew Flahive, Carlo Wouters, J. Wenny Rahayu, and David Taniar, 'Move: A distributed framework for materialized ontology view extraction', *Algorithmica*, **45**(3), 457–481, (2006).
[3] Mehul Bhatt, Joana Hois, and Oliver Kutz, 'Ontological Modelling of Form and Function in Architectural Design', *Applied Ontology Journal. IOS Press*, 1–35, (2012). (in press).
[4] Mehul Bhatt, J. Wenny Rahayu, Sury Prakash Soni, and Carlo Wouters, 'Ontology driven semantic profiling and retrieval in medical information systems', *J. Web Sem.*, **7**(4), 317–331, (2009).
[5] Mehul Bhatt, Carlo Wouters, Andrew Flahive, J. Wenny Rahayu, and David Taniar, 'Semantic completeness in sub-ontology extraction using distributed methods', in *ICCSA (3)*, pp. 508–517, (2004).
[6] P. Bouquet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krötzsch, S. Tessaris, D. Fensel, and A. Leger. Specification of a common framework for characterizing alignment, 2005. Knowledge Web Deliverable D2.2.1.
[7] M. Codescu and T. Mossakowski, 'Heterogeneous colimits', in *Proc. of MoVaH-08*, (2008).
[8] S. Coulson, *Semantic Leaps: Frame-Shifting and Conceptual Blending in Meaning Construction*, Cambridge University Press, 2001.
[9] J. Euzenat, 'Semantic precision and recall for ontology alignment evaluation', in *Proc. of IJCAI 2007*, ed., M. M. Veloso, pp. 348–353, (2007).
[10] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer, 2007.
[11] G. Fauconnier and M. Turner, *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*, Basic Books, 2003.
[12] K. Forbus, B. Falkenhainer, and D. Gentner, 'The structure-mapping engine', *Artificial Intelligence*, **41**, 1–63, (1989).
[13] P. Gärdenfors, *Conceptual Spaces - The Geometry of Thought*, Bradford Books, MIT Press, 2000.
[14] D. Gentner, 'Structure mapping: A theoretical framework for analogy', *Cognitive Science*, **7**(2), 155–170, (1983).
[15] J. A. Goguen, 'An Introduction to Algebraic Semiotics, with Applications to User Interface Design', in *Computation for Metaphors, Analogy and Agents*, number 1562 in LNCS, 242–291, Springer, (1999).
[16] J. A. Goguen and D. F. Harrell, 'Style: A Computational and Conceptual Blending-Based Approach', in *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*, Springer, (2009).
[17] J. A. Goguen and G. Malcolm, *Algebraic Semantics of Imperative Pro-*

grams, MIT, 1996.

[18] N. Guarino and C. Welty, 'Evaluating ontological decisions with OntoClean', *Commun. ACM*, **45**(2), 61–65, (2002).

[19] W. R. van Hage, *Evaluating Ontology-Alignment Techniques*, Ph.D. dissertation, Vrije Universiteit Amsterdam, 2008.

[20] J. Hois, O. Kutz, T. Mossakowski, and J. Bateman, 'Towards Ontological Blending', in *Proc. of the The 14th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA-2010)*, Varna, Bulgaria, September 8th–10th, (2010).

[21] M. Horridge and P. F. Patel-Schneider, 'Manchester Syntax for OWL 1.1', *OWLED-08*, (2008).

[22] I. Horrocks, O. Kutz, and U. Sattler, 'The Even More Irresistible $\mathcal{SROIQ}$', in *Proc. of KR*, eds., Patrick Doherty, John Mylopoulos, and Christopher A. Welty, pp. 57–67. AAAI Press, (2006).

[23] K. M. Jaszczolt, 'On Translating 'What Is Said': *Tertium Comparationis* in Contrastive Semantics and Pragmatics', in *Meaning Through Language Contrast Vol. 2*, 441–462, J. Benjamins, (2003).

[24] Y. Kalfoglou and M. Schorlemmer, 'Ontology mapping: the state of the art', *The Knowledge Engineering Review*, **18**(1), 1–31, (2003).

[25] W. Kuhn, 'Modeling the Semantics of Geographic Categories through Conceptual Integration', in *Proc. of GIScience 2002*, pp. 108–118. Springer, (2002).

[26] O. Kutz, D. Lücke, and T. Mossakowski, 'Heterogeneously Structured Ontologies—Integration, Connection, and Refinement', in *Proc. KROW 2008*, volume 90 of *CRPIT*, pp. 41–50. ACS, (2008).

[27] O. Kutz, D. Lücke, T. Mossakowski, and I. Normann, 'The OWL in the CASL—Designing Ontologies Across Logics', in *Proc. of OWLED-08*, volume 432. CEUR, (2008).

[28] O. Kutz, T. Mossakowski, and D. Lücke, 'Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design', *Logica Universalis*, **4**(2), 255–333, (2010). Special Issue on 'Is Logic Universal?'.

[29] O. Kutz and I. Normann, 'Context Discovery via Theory Interpretation', in *Workshop on Automated Reasoning about Context and Ontology Evolution, ARCOE-09 (IJCAI-09)*, (2009).

[30] M. Martinez, T. R. Besold, A. Abdel-Fattah, K.-U. Kühnberger, H. Gust, M. Schmidt, and U. Krumnack, 'Towards a Domain-Independent Computational Framework for Theory Blending', in *Proc. of the AAAI Fall 2011 Symposium on Advances in Cognitive Systems*, (2011).

[31] T. Mossakowski, C. Maeder, and K. Lüttich, 'The Heterogeneous Tool Set', in *TACAS*, volume 4424 of *LNCS*, pp. 519–522. Springer, (2007).

[32] Till Mossakowski, Christoph Lange, and Oliver Kutz, 'Three Semantics for the Core of the Distributed Ontology Language', in *7th International Conference on Formal Ontology in Information Systems (FOIS)*, ed., Michael Grüninger, Frontiers in Artificial Intelligence and Applications. IOS Press, (2012).

[33] I. Normann, *Automated Theory Interpretation*, Ph.D. dissertation, Jacobs University Bremen, 2009.

[34] R. E. Núñez, 'Creating mathematical infinities: Metaphor, blending, and the beauty of transfinite cardinals', *Journal of Pragmatics*, **37**, 1717–1741, (2005).

[35] F. C. Pereira, *Creativity and Artificial Intelligence: A Conceptual Blending Approach*, volume 4 of *Applications of Cognitive Linguistics (ACL)*, Mouton de Gruyter, Berlin, December 2007.

[36] G. D. Plotkin, 'A note on inductive generalization', *Machine Intelligence*, **5**, 153–163, (1970).

[37] D. Ritze, C. Meilicke, O. Šváb Zamazal, and H. Stuckenschmidt, 'A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences', in *OM-09*, volume 551 of *CEUR*, (2009).

[38] A. Schwering, U. Krumnack, K.-U. Kühnberger, and H. Gust, 'Syntactic Principles of Heuristic-Driven Theory Projection', *Cognitive Systems Research*, **10**(3), 251–269, (2009).

[39] M. Turner, 'The Way We Imagine', in *Imaginative Minds - Proc. of the British Academy*, ed., Ilona Roth, 213–236, OUP, Oxford, (2007).

[40] T. Veale, 'Creativity as pastiche: A computational treatment of metaphoric blends, with special reference to cinematic "borrowing"', in *Proc. of Mind II: Computational Models of Creative Cognition*, (1997).

[41] A. Zimmermann, M. Krötzsch, J. Euzenat, and P. Hitzler, 'Formalizing Ontology Alignment and its Operations with Category Theory', in *Proc. of FOIS-06*, pp. 277–288, (2006).