# Ontology Partitioning Using $\mathcal{E}$-Connections Revisited

## Extended Abstract

Sascha Jongebloed and Thomas Schneider*

Department of Computer Science, University of Bremen, Germany
{sasjonge,ts}@cs.uni-bremen.de

## 1 Introduction

Modular ontologies have received much attention in the past decade; they are usually easier to maintain, comprehend, and reason over. If an ontology is given as a monolithic entity, the task of *decomposing* it into modules is the first step towards making it modular. Several decomposition approaches have been developed, among them partitionings based on $\mathcal{E}$-connections [2] (henceforth: $\mathcal{E}$-partitions). $\mathcal{E}$-partitions have been developed for the purpose of automatically and efficiently decomposing an ontology into a graph whose nodes are (pairwise disjoint and mutually covering) *components* (i.e., subsets) of the ontology, and whose edges represent "semantic links" between the components in the style of $\mathcal{E}$-connections [5]. The adoption of the $\mathcal{E}$-connection framework ensures that the resulting partitions provide strong logical guarantees, such as encapsulation.

$\mathcal{E}$-Connections have been defined for *abstract description systems (ADSs)*, a notion that generalises description logics (DLs) and further formalisms. An $\mathcal{E}$-connection is a combination of (heterogeneous) logical theories via semantic links established by a designated set of relations, called *link relations*. The semantics of such a combination is given by interpretations consisting of pairwise disjoint components. In contrast to the general nature of $\mathcal{E}$-connections, $\mathcal{E}$-partitions have been defined specifically for the DL $\mathcal{SHOIQ}(\mathcal{D})$, a fragment of the latest OWL 2 ontology language. The partitioning procedure starts from a monolithic ontology $\mathcal{O}$ and attempts to turn it into an (as fine as possible) $\mathcal{E}$-connection by identifying link relations among the roles *in* $\mathcal{O}$. In [1,2] an efficient algorithm is given. To ensure that the resulting $\mathcal{E}$-partition and the input ontology are equivalent under the $\mathcal{E}$-connection semantics, an additional condition has to be imposed on the input ontology, which was called safety and coincides with *domain-independence (DI)*, as known from first-order logic and database theory. Contrary to DI in first-order logic, DI for $\mathcal{SHOIQ}(\mathcal{D})$ is decidable.

The partitioning algorithm was implemented as an experimental feature of the (discontinued) ontology editor Swoop. Initial experiments [2] showed that some ontologies admit useful $\mathcal{E}$-partitions, sometimes revealing modelling deficiencies. On the other hand, some modelling patterns – e.g., the use of few top-level concepts – are notoriously problematic: they admit only coarse $\mathcal{E}$-partitions. The limited success of $\mathcal{E}$-partitions may be due to this observation, but may also lie in the preliminary nature of the implementation. Hence some observed "poor" (i.e., coarse-grained) $\mathcal{E}$-partitions might be due to code bugs and not "features" of the general approach.

* Supported by DFG project SCHN 1234/3.

When pursuing this last question, we found a considerably simpler way of computing $\mathcal{E}$-partitions, based on the idea of creating an undirected graph $G$ whose edges connect concepts and/or roles that must be part of the same component, and reading the minimal $\mathcal{E}$-partition off $G$'s connected components. We were able to extend the underlying framework to most of OWL 2, with the only exception of the universal role $u$. This exception is unavoidable in some sense [3], but it is also insignificant because $u$ "typically plays a minor role in modelling" [4].

Our new approach offers the following advantages over the original one:

- Ready applicability to the latest OWL 2 standard, under two restrictions ensuring equivalence (domain-independence, absence of the universal role)
- A simplified notation of the theoretical foundations
- A new deterministic partitioning algorithm that is based on a simple idea and can be implemented to run in linear time (the original one is quadratic)
- Simple rigorous proofs that the algorithm is correct and outputs the maximal equivalent $\mathcal{E}$-connection

In the light of these advantages, we predict an easy implementation of the algorithm and plan experiments on an up-to-date corpus of existing ontologies. The work reported here is therefore in progress. This extended abstract summarises the submission [3] to the Description Logic Workshop, which contains numerous additional details.

## 2 $\mathcal{E}$-Connections and Partitionings for $\mathcal{SROIQ}$

We consider $\mathcal{SROIQ}$, the description logic (DL) underlying OWL, minus the universal role (see above). We omit datatypes and keys, discussing their addition (and of further non-OWL features) in [3]. For the syntax and semantics of $\mathcal{SROIQ}$, see [4]. An *ontology* is a set of axioms (no distinction between TBoxes, RBoxes, and ABoxes).

Let $\Sigma$ be a *signature*, i.e., a finite set of *terms* (concept, role, and individual names). Given a natural number $n \geq 1$, an *$n$-numbering of $\Sigma$* is a function $\nu$ that assigns to each concept and individual name a number $\nu(A), \nu(a) \in \{1, \ldots, n\}$ and to each role name a number $\nu(r) \in \{1, \ldots, n\}^2$. Numberings are extended inductively to arbitrary concepts and axioms. A concept $C$ (axiom $\alpha$) is called an *$i$-concept ($i$-axiom)* if $\nu(C) = i$ ($\nu(\alpha) = i$).

Given an $n$-numbering $\nu$, a *$\nu$-ontology* is a tuple $\mathbb{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$, each of whose *components* $\mathcal{O}_i$ is a nonempty set of $i$-axioms. The semantics of $\nu$-ontologies is given by $\nu$-interpretations, whose domain consist of $n$ pairwise disjoint nonempty sets, and which interpret every $i$-concept name and $i$-individual name in the $i$-th component, and every $(i, j)$-role name as a relation between the $i$-th and $j$-th component. The interpretation function is extended in the obvious way to arbitrary concepts; satisfaction of $i$-axioms is defined as expected and denoted $\mathcal{I} \models^{\nu} \alpha$. A $\nu$-ontology $\mathcal{I}$ is a *model* of a $\nu$-ontology $\mathbb{O}$, written $\mathcal{I} \models^{\nu} \mathbb{O}$, if $\mathcal{I} \models^{\nu} \alpha$ for all axioms $\alpha$ in $\mathbb{O}$. $\mathbb{O}$ is *consistent* if it has a model.

The correspondence between simple and $\nu$-ontologies is captured by compatibility and equivalence (the former being syntactic and the latter semantic).

**Definition 1.** Let $\mathcal{O}$ be an ontology, $\nu$ an $n$-numbering, and $\mathbb{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ a $\nu$-ontol.

1. $\mathcal{O}$ and $\mathbb{O}$ are *compatible*, written $\mathcal{O} \sim \mathbb{O}$, if $\mathcal{O} = \bigcup_{i \leq n} \mathcal{O}_i$ .
2. $\mathcal{O}$ and $\mathbb{O}$ are *equivalent*, written $\mathcal{O} \approx \mathbb{O}$ if, for all $\nu$-interpret. $\mathcal{I}$: $\mathcal{I} \models \mathcal{O}$ iff $\mathcal{I} \models^{\nu} \mathbb{O}$.

Compatibility implies equivalence under an additional assumption: domain-independence as known from the first-order and database worlds. A concept $C$ (axiom $\alpha$) is *domain-independent (DI)* if $C^{\mathcal{I}} = C^{\mathcal{J}}$ ($\mathcal{I} \models \alpha$ iff $\mathcal{J} \models \alpha$) for all interpretations $\mathcal{I}, \mathcal{J}$ with $X^{\mathcal{I}} = X^{\mathcal{J}}$ for all terms $X$. An ontology is DI if so are all its axioms. For (most of) $\mathcal{SROIQ}$, DI can be decided efficiently via a syntactic characterisation, called *locality* in [1,2]. DI links compatibility and equivalence as follows.

**Theorem 2.** *Let $\mathcal{O}$ be an ontology, $\nu$ an n-numbering, $\mathbb{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ a $\nu$-ontology.*

1. *(a) If $\mathcal{O}$ is DI and $\mathcal{O} \sim \mathbb{O}$, then $\mathcal{O} \approx \mathbb{O}$.*
   *(b) If additionally $\mathcal{O}$ is consistent, then so is $\mathbb{O}$.*
2. *If $\mathcal{O}$ is not DI and consistent and $\mathcal{O} \sim \mathbb{O}$ and $\mathcal{O} \approx \mathbb{O}$, then $n = 1$, i.e., $\mathbb{O} = \mathcal{O}$.*

## 3 The New Partitioning Algorithm

We now present the partitioning algorithm. As in [1,2], it receives as input an ontology $\mathcal{O}$ and returns a $\nu$-ontology $\mathbb{O} = (\mathcal{O}_1, \ldots, \mathcal{O}_n)$ such that $\mathcal{O} \sim \mathbb{O}$ and $n$ is maximal with this property. If $\mathcal{O}$ is domain-independent, $\mathcal{O} \approx \mathbb{O}$ follows by Theorem 2. Let $\mathsf{sub}(\mathcal{O})$ be the set of all concepts (atomic or complex) occurring in $\mathcal{O}$. The main routine `partition(`$\mathcal{O}$`)` of Algorithm 1 first creates a graph $G$ containing one node per concept in $\mathsf{sub}(\mathcal{O})$ and two nodes $r_0, r_1$ per role $r$ in $\mathcal{O}$. It then adds all edges induced by the structure of the concepts (`addSubConceptEdges`) and axioms (`addAxiomEdges`) in $\mathcal{O}$ to $G$. For a given role $R$, both subroutines use the notation $R_i$, which equals $r_i$ if $R = r$ and $r_{1-i}$ if $R = r^-$, for $i = 0, 1$. Additionally, `addAxiomEdges` labels, for each axiom $\alpha$, one of the created edges with $\alpha$. Next, the CCs of $G$ are determined. Then the partitioning is read off the axiom labels in the CCs. We show in [3] that the algorithm runs in linear time, is correct, and outputs the maximal compatible $\mathbb{O}$.

For example, let $\mathcal{O} = \{A \sqsubseteq \exists r.B, B \sqsubseteq B'\}$. Then $G$ has nodes $A, \exists r.B, r_0, r_1, B, B'$, and edges $\{A, \exists r.B\}$, $\{\exists r.B, r_0\}$, $\{r_1, B\}$, $\{B, B'\}$. Edges $\{A, \exists r.B\}$ and $\{B, B'\}$ are labelled $A \sqsubseteq \exists r.B$ and $B \sqsubseteq B'$, respectively. Now $G$ has 2 connected components (CCs): $G_1$ with nodes $A, \exists r.B, r_0$ and label $\alpha$; $G_2$ with $r_1, B, B'$ and $\beta$. Hence $\mathbb{O} = (\{A \sqsubseteq \exists r.B\}, \{B \sqsubseteq B'\})$.

## 4 Conclusions and Future Work

We have extended the original approach underlying $\mathcal{E}$-partitions in [1,2] to all of OWL 2 except the universal role (which cannot be accommodated, as shown in [3]). We have presented a new simplified notation and a linear-time algorithm for computing the maximal $\mathcal{E}$-connection that is syntactically compatible (and, assuming domain-independence, equivalent) with the input ontology. We show in [3] that theory and algorithm extend to expressive operators on roles considered in the literature.

For future work, we expect a straightforward implementation of our algorithm, as a basis for experiments on a representative up-to-date ontology corpus. We conjecture that existing ontologies generally decompose well when allowing slight deviations from (syntactic) compatibility, to circumvent the notorious problematic modelling patterns, see §1. We furthermore plan to revisit module extraction, extending the existing procedure

**Algorithm 1:** Partitioning an ontology $\mathcal{O}$

**Function** `partition`$(\mathcal{O})$**:**
  **input :** $\mathcal{O}$ with signature $\Sigma$ **output:** $\nu$-ontology $\mathbb{O}$
  $V \leftarrow \{C \mid C \in \mathsf{sub}(\mathcal{O})\} \cup \{r_0, r_1 \mid r \in \Sigma_R\}$; $E \leftarrow \emptyset$; $L \leftarrow \emptyset$
  **forall** $C \in \mathsf{sub}(\mathcal{O})$ **do** `addSubConceptEdges`$(G, C)$
  **forall** $\alpha \in \mathcal{O}$ **do** `addAxiomEdges`$(G, \alpha)$
  $\{G_1, \ldots, G_n\} \leftarrow$ all connected components (CCs) of $G$ with $\geq 1$ axiom label
  **forall** $i \leq n$ **do** $\mathcal{O}_i \leftarrow \{\alpha \mid L(v, v') = \alpha$ for some edge $(v, v')$ in $G_i\}$
  $\mathbb{O} \leftarrow (\mathcal{O}_1, \ldots, \mathcal{O}_n)$
  **return** $(\mathbb{O})$

**Function** `addSubConceptEdges`$(G, C)$**:**
  **switch** $C$ **do**
    **case** $\neg D$ **do** $E \leftarrow E \cup \{C, D\}$
    **case** $D \sqcap F$ **do** $E \leftarrow E \cup \{\{C, D\}, \{C, F\}\}$
    **case** $\geqslant m\, R.D$ **do** $E \leftarrow E \cup \{\{C, R_0\}, \{R_1, D\}\}$
    **case** $\exists R.\mathsf{Self}$ **do** $E \leftarrow E \cup \{\{C, R_0\}, \{C, R_1\}\}$
    **case** $\{a\}$ **do** $E \leftarrow E \cup \{\{C, a\}\}$

**Function** `addAxiomEdges`$(G, \alpha)$**:**
  **switch** $\alpha$ **do**
    **case** $C \sqsubseteq D$ *or* $C \equiv D$ **do** $E \leftarrow E \cup \{C, D\}$; $L(C, D) \leftarrow \alpha$
    **case** $R \sqsubseteq S$, $R \equiv S$ *or* $\mathsf{Disjoint}(R, S)$ **do**
      $E \leftarrow E \cup \{\{R_0, S_0\}, \{R_1, S_1\}\}$; $L(R_0, S_0) \leftarrow \alpha$
    **case** $R \circ S \sqsubseteq T$ **do** $E \leftarrow E \cup \{\{R_1, S_0\}, \{R_0, T_0\}, \{S_1, T_1\}\}$; $L(R_0, T_0) \leftarrow \alpha$
    **case** $C(a)$ **do** $E \leftarrow E \cup \{C, a\}$; $L(C, a) \leftarrow \alpha$
    **case** $R(a, b)$ **do** $E \leftarrow E \cup \{\{a, R_0\}, \{R_1, b\}\}$; $L(a, R_0) \leftarrow \alpha$
    **case** $a \approx b$ *or* $a \not\approx b$ **do** $E \leftarrow E \cup \{\{a, b\}\}$; $L(a, b) \leftarrow \alpha$

in [2] to arbitrary $\mathcal{E}$-connections, independently of a specific partitioning algorithm. Given the linear runtime of our algorithm, module extraction might even compete in performance with syntactic locality. Finally, we expect to transfer the overall approach to logics with unary negation, such as frontier-one existential rules or even UNFO.

## References


1. Cuenca Grau, B.: Combination and integration of ontologies on the semantic web. Ph.D. thesis, Universidad de Valencia (2005)
2. Cuenca Grau, B., Parsia, B., Sirin, E., Kalyanpur, A.: Modularity and web ontologies. In: Proc. of KR-06. pp. 198–209. AAAI Press (2006)
3. Jongebloed, S., Schneider, T.: Ontology partitioning using $\mathcal{E}$-connections revisited (2018), DL 2018. TR: `http://www.informatik.uni-bremen.de/tdki/research/papers.html`
4. Krötzsch, M., Simančik, F., Horrocks, I.: A description logic primer. CoRR **abs/1201.4089** (2012), `http://arxiv.org/abs/1201.4089`
5. Kutz, O., Lutz, C., Wolter, F., Zakharyaschev, M.: $\mathcal{E}$-connections of abstract description systems. Artif. Intell. **156**(1), 1–73 (2004)