Relations between Ontology Properties

Siegfried Bublitz
Siemens IT Solutions and Services GmbH

C-LAB Report

Vol. 10 (2011) No. 02

Cooperative Computing & Communication Laboratory

ISSN 1619-7879

C-LAB ist eine Kooperation
der Universität Paderborn und der Siemens IT Solutions and Services GmbH
www.c-lab.de
info@c-lab.de

C-LAB Report

Herausgegeben von
Published by

Dr. Wolfgang Kern, Siemens IT Solutions and Services GmbH
Prof. Dr. Franz-Josef Rammig, Universität Paderborn

Das C-LAB - Cooperative Computing & Communication Laboratory - leistet Forschungs- und Entwicklungsarbeiten und gewährleistet deren Transfer an den Markt. Es wurde 1985 von den Partnern Nixdorf Computer AG (nun Siemens IT Solutions and Services GmbH) und der Universität Paderborn im Einvernehmen mit dem Land Nordrhein-Westfalen gegründet.

Die Vision, die dem C-LAB zugrunde liegt, geht davon aus, dass die gewaltigen Herausforderungen beim Übergang in die kommende Informationsgesellschaft nur durch globale Kooperation und in tiefer Verzahnung von Theorie und Praxis gelöst werden können. Im C-LAB arbeiten deshalb Mitarbeiter von Hochschule und Industrie unter einem Dach in einer gemeinsamen Organisation an gemeinsamen Projekten mit internationalen Partnern eng zusammen.

C-LAB - the Cooperative Computing & Cooperation Laboratory - works in the area of research and development and safeguards its transfer into the market. It was founded in 1985 by Nixdorf Computer AG (now Siemens IT Solutions and Services GmbH) and the University of Paderborn under the auspices of the State of North-Rhine Westphalia.

C-LAB's vision is based on the fundamental premise that the gargantuan challenges thrown up by the transition to a future information society can only be met through global cooperation and deep interworking of theory and practice. This is why, under one roof, staff from the university and from industry cooperate closely on joint projects within a common research and development organization together with international partners. In doing so, C-LAB concentrates on those innovative subject areas in which cooperation is expected to bear particular fruit for the partners and their general well-being.

ISSN 1619-7879

C-LAB
Fürstenallee 11
33102 Paderborn

fon: +49 5251 60 60 60
fax: +49 5251 60 60 66
email: info@c-lab.de
Internet: www.c-lab.de

Abstract

Ontologies provide a means for modelling knowledge in computer processable form and properties play a fundamental role when designing ontologies. For fine-tuning knowledge, different kinds (or attributes) of properties are available. These kinds are defined by first order logic, allowing ontology reasoning tools to automatically generate additional knowledge which can be investigated by applications deploying ontology reasoners. This means the generated knowledge heavily depends on the kind of properties used in ontology modelling. In this paper we investigate how these property kinds differ, how they relate to each other and how this can be depicted graphically. Even so most of the results are quite obvious, we use a formal approach to deduce them exactly.

Contents

1 Introduction and Notation....4

1.1 Notation....4

1.2 Kinds of Properties in OWL....5

2 Kinds of properties....6

2.1 Functional properties....6

2.2 Inverse Functional properties....7

2.3 Symmetric properties....7

2.4 Antisymmetric properties....7

2.5 Asymmetric properties....8

2.6 Reflexive properties....8

2.7 Irreflexive properties....8

2.8 Transitivity....9

2.8.1 The Transitive Hull....9

2.8.2 Graphical Construction of Transitivity....9

2.9 Simple Examples of Properties....10

2.10 Observations on properties 11

3 Conclusions....15

4 References....16

1 Introduction and Notation

Ontologies describe the domain of concept of a problem area by using classes, objects and properties. Objects are instances of classes. Properties are binary relations combining classes with classes thus providing properties between objects or combining classes with values thus providing properties between objects and values. A prominent language elaborating these concepts is OWL [1].

The properties can be specialized and of different kinds so to fulfill certain requirements which are well known from binary relationships within mathematics. These kinds of properties are defined by the appropriate OMG standard and realized in editing environments as e.g. Protégé [2]. Handling OWL is quite ambitious, Bechhofer et. al. complain in [3] about „subtly different ways and confusion reigns“ when application developer interpret language specifications.

To get a deeper feeling for this properties and how they can be combined, a graphical representation is often very helpful. In this paper we transmit the logical restrictions induced by the different types of properties and show how this affects their graphical representation. The results of this work are used for ontology modelling and incorporation in up-to-date research projects like OEPI [4].

1.1 Notation

If M is the set of all classes – or classes and value sets – used in the ontology, a property R is just a subset of the Cartesian product of M with itself:

$$R \subset M \times M$$

For a pair $(i, j) \in R$ we say i in the *domain* of R is *associated* with j in the *range* of R .

Basically, there is no great difference when observing properties on $M \times N$ for different sets M and N: Just substitute each component with the union of M and N and we again have a property on identical component sets. This allows us not to have to distinguish explicitly properties between classes and classes, or between classes and values, or between objects, etc.

As we are dealing with finite sets, we can identify M with a subset of the natural numbers, that is, when $n \in \mathbb{N}$ is the *size* of M, we identify $M = \{1, \ldots, n\}$. This allows us to present R as a binary matrix, $(r_{ij}), 1 \leq i, j \leq n$ called the *adjacency matrix* of R with the definition:

$$r_{ij} = 1 \ \ if (i, j) \in R; \quad r_{ij} = 0 \ \ otherwise$$

Additionally, we have the ordering properties induced by natural numbers available in M. In the following, we do not explicitly distinct between R and the corresponding adjacency matrix as we can obviously construct one from the other and thus just write $R=(r_{ij})$. Formally, we have a bijection

$$R: M \times M \rightarrow \{0,1\}^{n*n}$$

which allows us to use matrix operations or set operations, whatever is more appropriate.

Let S be another property on M. The following operations known from elementary set theory are listed here just to agree upon notation:

- *complement* $\overline{R}=\{(i,j)\in M\times M \mid (i,j)\notin R\}$
- *union* $R\cup S=\{(i,j)\in M\times M \mid (i,j)\in R\vee(i,j)\in S\}$
- *intersection* $R\cap S=\{(i,j)\in M\times M \mid (i,j)\in R\wedge(i,j)\in S\}$
- *difference* $R\setminus S=\{(i,j)\in M\times M \mid (i,j)\in R\wedge(i,j)\notin S\}$
- the *size* of the set R, denoted $|R|$, is the number of elements contained in the set R.

Apart from this, the matrix structure allows the following definitions:

- *inverse (mirrored, transposed*) property $R^{-1}=\{(i,j) \mid (j,i)\in R\}$
- *main diagonal* $D=\{(i,i) \mid i\in M\}$
- row (horizontal line) at i: $H(i)=\{(i,j) \mid j\in M\}$
- column (vertical line) at j: $V(j)=\{(i,j) \mid i\in M\}$

1.2 Kinds of Properties in OWL

The following types of properties are defined in the OMG OWL2 specification:

- *functional*, i.e. each domain entry is associated to at most one range entry.
- *inverse functional*, i.e. each range entry is associated to at most one domain entry.
- *symmetric*, i.e. if one entry is associated to another, the second is associated to the first one as well.

- *antisymmetric*, two different entities can not be associated in both directions simultaneously.
- *asymmetric*, i.e. if one entry is associated to another, the second is not associated to the first.
- *reflexive*, i.e. each entity is associated to itself.
- *irreflexive*, i.e. no entity is associated to itself.
- *transitive*, i.e. if one entry is associated to a second one and the second one associated to a third one, the first is associated to the third as well.

Having specified a property in an OWL ontology - e.g. in an ontology editor like Protégé [2] - and stated the property to be of one of the specified kinds means a reasoning tool will create associations (entries) which are demanded by this kind of property. Simultaneously, the reasoning tool will warn when entries are created which are prohibited by this kind of property. The better understanding of the effects of modelling with different kinds of properties is a central topic of this paper. This helps to understand conclusions computed by the reasoner.

2 Kinds of properties

In this chapter we examine the kinds of properties used in OWL in more detail. We formulate the conditions stated in OWL in logical and set theoretical terms and give a graphical interpretation as well.

2.1 Functional properties

A functional property can be considered as a functional mapping from the domain to the range. Logically this means if two pairs are identical in the *first* argument, they are identical in the *second* argument as well:

$$(i,\ j)\in R \wedge (i\ ,k)\in R \Rightarrow j=k.$$

or correspondingly: $|R\cap H(i)| \leq 1\ \forall\ i\in M.$

If R is *total functional*, i. e. for every element $i\in M$ there is $j\in M$ with $(i,\ j)\in R,$ we can write R as a function $R: M\rightarrow M.$

Graphical condition: Each row in the adjacency matrix ($H(i)$ in the picture) contains at most one entry (with value 1).

$$\begin{matrix} & & j & & \\ & & \vdots & & \\ i & \cdots & 1 & \cdots & H(i) \\ & & \vdots & & \end{matrix}$$

If R is total functional, each row contains exactly one entry.

Observation: The intersection of functional properties is functional as well.

2.2 Inverse Functional properties

For an inverse functional property the inverse property is functional. This can be considered as a functional mapping from the domain to the range. Logically this means if two pairs are identical in the *second* argument, they are identical in the *first* argument as well:

$$(i,\ j)\in R\wedge(h,\ j)\in R\Rightarrow i=h.$$

or correspondingly: $|R\cap V(j)|\ \leq 1\ \forall\ j\in M.$

If R^{-1} is *total functional*, i.e. for every element $j\in M$ there is $i\in M$ with $(i,\ j)\in R,$ we can write R^{-1} as a function $R^{-1}: M\rightarrow M.$

$$\begin{array}{cccc} & & j & \\ & & \vdots & \\ i & \cdots & 1 & \cdots \\ & & \vdots & \\ & & V(j) & \end{array}$$

Graphical condition: Each column in the adjacency matrix ($V(j)$ in the picture) contains at most one entry (with value 1).

If R^{-1} is total functional, each column contains exactly one entry.

Observation: The intersection of inverse functional properties is inverse functional as well.

2.3 Symmetric properties

The logical condition for symmetry is

$$(i,\ j)\in R\Rightarrow(j,i)\in R$$

or correspondingly: $R^{-1}=R.$

Graphical condition: The adjacency matrix is mirrored at the main diagonal.

Observation: The intersection of symmetric properties is symmetric as well.

$$\begin{array}{ccccc} & & i & \cdots & j \\ & \ddots & \vdots & & \vdots \\ i & \cdots & \ddots & \cdots & 1 \\ \vdots & & \vdots & \ddots & \\ j & \cdots & 1 & & \end{array}$$

2.4 Antisymmetric properties

The logical condition for antisymmetry is

$$(i,\ j)\in R\wedge(j,i)\in R\Rightarrow i=j$$

or correspondingly: $R^{-1}\cap R\subset D.$

Graphical condition: There are no mirrored entries allowed, except on the main diagonal.

$$\begin{array}{ccccc} & & i & \cdots & j \\ & \ddots & \vdots & & \vdots \\ i & \cdots & ? & \cdots & 1 \\ \vdots & & \vdots & \ddots & \\ j & \cdots & 0 & & \end{array}$$

Observation: The intersection of antisymmetric properties is antisymmetric as well.

2.5 Asymmetric properties

The logical condition for asymmetry is

$$(i,\ j) \in R \Rightarrow (j,i) \notin R$$

or correspondingly: $R^{-1} \cap R = \emptyset$.

Graphical condition: There are no mirrored entries allowed, not even on the main diagonal.

$$\begin{array}{ccccc} & & i & \cdots & j \\ & \ddots & \vdots & & \vdots \\ i & \cdots & 0 & \cdots & 1 \\ \vdots & & \vdots & \ddots & \\ j & \cdots & 0 & & \end{array}$$

Observation: The intersection of asymmetric properties is asymmetric as well.

2.6 Reflexive properties

The logical condition for reflexivity is

$$i \in M \Rightarrow (i,i) \in R$$

or correspondingly: $D \subset R.$

Graphical condition: The main diagonal is contained in the property.

$$\begin{array}{ccc} 1 & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & 1 \end{array}$$

Observation: The intersection of reflexive properties is reflexive as well.

2.7 Irreflexive properties

The logical condition for irreflexivity is

$$i \in M \Rightarrow (i,i) \notin R$$

or correspondingly: $D \cap R = \emptyset$

Graphical condition: Every point in the main diagonal is excluded from the property.

$$\begin{matrix} 0 & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & 0 \end{matrix}$$

Observation: The intersection of irreflexive properties is irreflexive as well.

2.8 Transitivity

Transitivity is more complex than the other kinds, so it is discussed here in more detail.

The logical condition for transitivity is

$$(i, j) \in R \wedge (j, k) \in R \Rightarrow (i, k) \in R.$$

2.8.1 The Transitive Hull

Each property R can be extended by putting additional entries into it which are demanded by the transitivity condition. This can be continued until the resulting property is transitive. As the example with the complete property $M \times M$ shows, this is always possible. Furthermore, the result is independent from the insertion order of new entries. Therefore, the result is unique and is called the *transitive hull* of the property, noted as R^* and formalized as the smallest transitive property that contains the original one and is transitive. Since the intersection of two transitive properties is transitive as well, we can define

$$R^* = \cap \{S | S \, transitive, R \subset S\}$$

Observation 1: $R \subset R^*$.

Observation 2: R is transitive if and only if $R = R^*$.

2.8.2 Graphical Construction of Transitivity

The graphical aspects of transitivity are depicted by the following picture. The construction can be used to create the transitive hull of a given property.

$$\begin{array}{cccccccc}
 & & & j & \cdots & k & & \\
 & \ddots & & \vdots & & \vdots & & \\
i & \cdots \ddots \cdots & & 1 & \cdots & \underline{1} & \cdots & H(i) \\
\vdots & & \ddots & \vdots & & \vdots & & \uparrow \\
j & \cdots \cdots \cdots & & * & \cdots & 1 & \cdots & H(j) \\
 & & & \vdots & \ddots & & & \\
 & & & V(j) & & & &
\end{array}$$

Graphically, the following is happening: Consider the point (i, j) marked with 1 in the matrix. This point lies on the column $V(j)$. The point where $V(j)$ intersects the main diagonal is marked with $*$ in the matrix. Consider the row $H(j)$ which contains this point. Every entry marked 1 in $H(j)$ corresponds to an entry (j, k) in R and transitivity requires an entry (i, k) to exist, which is just an entry marked 1 in $H(i)$ at position k. We can say for the property R: *every entry in* $H(j)$ ***induces*** *the existence of an entry in* $H(i)$ *in the same column*. More suggestively: $H(j)$ ***projects*** *onto* $H(i)$, which is illustrated by the vertical arrow in the picture.

Observation 3: $H(i)$ contains at least as much 1's as $H(j)$: $|R \cap H(j)| \leq |R \cap H(i)|$

Observation 4: If $(j, i) \in R$ then $H(i)$ projects on $H(j)$ as well. This means there is the same number of 1's at the same columns in $H(i)$ and $H(j)$.

2.9 Simple Examples of Properties

Here are some examples of simple properties. The effects are illustrated in Table 1.

- Complete property $R = M \times M$
- Empty property $R = \emptyset$
- Main Diagonal $D = \{(i, i) \mid i \in M\}$
- Row (horizontal line) at i: $H(i) = \{(i, j) \mid j \in M\}$
- Column (vertical line) at j: $V(j) = \{(i, j) \mid i \in M\}$

R	Reflexive	Irreflexive	Symmetric	Asymmetric	Antisymmetric	Transitive	Functional	Inverse Functional
$M \times M$	+		+			+		
$\emptyset$		+	+	+	+	+	+	+
D	+		+	+	+	+	+	+
$H(i)$				+	+	+		+
$V(j)$				+	+	+	+	

Table 1: Attributes of example properties

2.10 Observations on properties

Observation on reverse property: If R is reflexive, irreflexive, symmetric, asymmetric, antisymmetric, or transitive than R^{-1} is so as well. If R is functional than R^{-1} is inverse functional and vice versa.

Proof:

a) Assume R is reflexive. Then $D \subset R$ and definition of D gives $D \subset R^{-1}$, thus R^{-1} is reflexive.

b) Assume R is irreflexive. Then $D \subset R$ and definition of D gives $D \cap R^{-1} = \emptyset$, thus R^{-1} is irreflexive.

c) Assume R is symmetric. Let $(i, j) \in R^{-1}$. Then $(j, i) \in R$. Symmetry of R requires $(i, j) \in R$, which means $(j, i) \in R^{-1}$, thus R^{-1} is symmetric.

d) Assume R is asymmetric. Let $(i, j) \in R^{-1}$. Then $(j, i) \in R$. Asymmetry of R requires $(i, j) \notin R$, which means $(j, i) \notin R^{-1}$, thus R^{-1} is asymmetric.

e) Assume R is antisymmetric. Let $(i, j) \in R^{-1} \wedge (j, i) \in R^{-1}$. Then $(j, i) \in R \wedge (i, j) \in R$. Antisymmetry of R requires $i = j$, thus R^{-1} is antisymmetric.

f) Assume R is transitive. Let $(i, j), (j, k) \in R^{-1}$. Then $(j, i), (k, j) \in R$. Transitivity of R requires $(k, i) \in R$, which means $(i, k) \in R^{-1}$, thus R^{-1} is transitive.

g) Assume R is functional. Let $(i, j) \in R^{-1} \wedge (h, j) \in R^{-1}$. Then $(j, i) \in R \wedge (j, h) \in R$. Functionality of R requires $i = h$, thus R^{-1} is inverse functional.

h) Assume R is inverse functional. Let $(i, j) \in R^{-1} \wedge (i, k) \in R^{-1}$. Then $(j, i) \in R \wedge (k, i) \in R$. Inverse functionality of R requires $j = k$, thus R^{-1} is functional□

The next observations show certain kinds of property combinations to result in a single property, the Diagonal.

Observation on reflexivity: If R is reflexive and functional or inverse functional, than $R = D$.

Proof: Let $(i, j) \in R$. Because of reflexivity we have $(i, i) \in R$ and each of functionality or inverse functionality requires $i = j$ which means $(i, i) \in R$, thus $R \subset D$. On the other hand reflexivity requires $D \subset R$, thus $R = D$. □

Observation on transitivity: If R is total functional, inverse functional and transitive than $R = D$.

Proof: Let $(i, j) \in R$. R being total requires there is $k \in M$ with $(j, k) \in R$. Transitivity of R requires $(i, k) \in R$. Functionality of R requires $j = k$. Thus $(i, j) \in R \wedge (j, j) \in R$. Inverse functionality of R requires $i = j$, thus $(i, j) \in D$. This means $R \subset D$.

On the other hand let $(i, i) \in D$. R being total requires there is $j \in M$ with $(i, j) \in R$. Simultaneously as above we conclude $i = j$, and thus $(i, i) \in R$. Thus $D \subset R$. Together this means $R = D$ □

Another combination of kinds or properties gives only parts of the Diagonal.

Observation on symmetry: If R is symmetric and antisymmetric then $R \subset D$.

Proof: Let $(i,j)\in R.$ Because of symmetry we have $(j,i)\in R$ and antisymmetry requires $i=j$ which means $(i,j)\in D,$ thus $R\subset D.$ □

A *permutation matrix* is a matrix having exactly one “1”-entry in each row and each column.

Observation on functionality: If R is functional and inverse functional the corresponding matrix is a permutation matrix.

Proof: Obvious from conditions.□

If R is functional and inverse functional it is possible to rearrange the ordering of elements in M until the corresponding matrix is a *diagonal matrix* where exactly the diagonal entries (i,j) correspond to 1. Formally, this means there is a bijection $b:M\rightarrow M$ with $R=\{(i,b(i))\mid i\in M\}$ This might be useful but of course has effects on other properties of the same set.

The following observations show that there is only one property fulfilling certain combinations of conditions, namely the empty property.

Observation on asymmetry: If R is symmetric and asymmetric then $R=\emptyset$.

Proof: Let $(i,j)\in R.$ Because of symmetry we have $(j,i)\in R.$ On the other hand asymmetry require $(j,i)\notin R,$ which is a contradiction. Thus there is no $(i,j)\in R,$ which means $R=\emptyset$ □

Observation on antisymmetry: R is asymmetric exactly when R is irreflexive and antisymmetric.

Proof: Assume R is asymmetric. Let $(i,j)\in R.$ Asymmetry of R requires $(j,i)\notin R.$ Thus $i\neq j,$ which means R is irreflexive. Furthermore, the condition $(i,j)\in R\wedge(j,i)\in R$ is never true, thus the conclusion $i=j$ is always correct, thus R is antisymmetric.

Now assume R is irreflexive and antisymmetric. Let $(i,j)\in R.$ Irreflexivity of R requires $i\neq j.$ Antisymmetry of R requires $(j,i)\notin R.$ Thus R is asymmetric□

R	**Logical Condition**	**Set Theoretical Condition**	**Graphical Condition**	R^{-1}
Reflexive	$(i,i) \in R$	$D \subset R$	Contains the main diagonal	Reflexive
Irreflexive	$(i,i) \notin R$	$D \cap R = \emptyset$	Disjoint with the main diagonal	Irreflexive
Symmetric	$(i,j) \in R \Rightarrow (j,i) \in R$	$R^{-1} = R$	Mirrored at main diagonal	Symmetric
Asymmetric	$(i,j) \in R \Rightarrow (j,i) \notin R$	$R^{-1} \cap R = \emptyset$	No mirrored elements allowed, not even on main diagonal	Asymmetric
Antisymmetric	$(i,j),(j,i) \in R \Rightarrow i = j$	$R^{-1} \cap R \subset D$	No mirrored elements allowed, except on main diagonal	Antisymmetric
Transitive	$(i,j),(j,k) \in R \Rightarrow (i,k) \in R$		$H(j)$ projects onto $H(i)$	Transitive
Functional	$(i,j),(i,k) \in R \Rightarrow j = k$	$\lvert R \cap H(i) \rvert \leq 1$	Each row contains at most one entry	Inverse functional
Inverse Functional	$(i,j),(j,k) \in R \Rightarrow i = h$	$\lvert R \cap V(j) \rvert \leq 1$	Each column contains at most one entry	Functional

Table 2: Attributes of property R *in logical, set theoretical and graphical description and effect on reverse property* R^{-1}

	Reflexive	Irreflexive	Symmetric	Asymmetric	Antisymmetric	Transitive	Functional	Inverse Functional
Reflexive		impossible		impossible			$R=D$	$R=D$
Irreflexive					asymmetric			
Symmetric				$R=\emptyset$	$R\subset D$			
Asymmetric								
Antisymmetric								
Transitive								
Functional								Permutation Matrix
Inverse Functional								

Table 3: Effects of combined attributes of property R

3 Conclusions

A good understanding of properties is a fundamental knowledge required of ontology developers. Here we used elementary set theory and graphics to give a foundation for this.

The effects of special kinds of properties on other properties and the combination of different properties have effects on the knowledge gained by the reasoner when exploiting this properties. Understanding and foreseeing these effects is the result of this paper and helps in developing consistent ontologies.

4 References

[1] *OWL 2 Web Ontology Language, Document Overview,* W3C, W3C Recommendation 27 October 2009, http://www.w3.org/TR/2009/REC-owl2-overview-20091027/

[2] Holger Knublauch, *Ontology-Driven Software Development in the Context of the Semantic Web: An Example Scenario with Protegé/ OWL,* Stanford Medical Informatics, Stanford University, CA

[3] Sean Bechhofer, Raphael Volz, Phillip W. Lord. *Cooking the Semantic Web with the OWL API.* In Proceedings of International Semantic Web Conference'2003. pp.659~675

[4] http://www.oepi-project.eu