Description Logic Workshop 1994, Bonn D, May 1994.

A Relation-Based Description Logic

Enrico Franconi and **Vania Rabito**
Knowledge Representation and Reasoning Lab.
IRST, I-38050 Povo TN, Italy
{franconi, rabito}@irst.it

1 Introduction

In this paper, we will show the basic ideas for a possible formalization[1] of KODIAK, a relation-based description logic, first informally introduced by Robert Wilensky in [Wilensky,1987]. His work was motivated by particular requirements of natural language applications, specially regarding lexical semantic discrimination: whenever the syntactical parser tries to build a constituent, the consistency of the semantic part of such a constituent is checked, and a minimal semantic representation is computed [Norvig,1987; Jacobs,1992; Bayer and Vilain,1991]. In [Wilensky,1987] several drawbacks of Kl-One-based languages were highlighted. The main comment is about the limited expressive power of Kl-One-based languages to represent *relational entities*. They are represented as *roles*, which are second-class citizens:

- Role restrictions do not get defined. E.g., `PAINTER` is defined via a relation `paints` restricted to `PAINTING`; however, the concept of `PAINTING` is left primitive. Inverse roles solve this problem; however, cyclic definitions are needed.
- Same relational entities need to be defined twice, as roles – in order to define concepts they are related to –and as concepts – in order to better define the relational entities themselves. However, no connection exists between the relation as role and the relation as concept.

The paper is organized as follows. After an informal introduction to KODIAK, we describe $\mathcal{RDL}$, a formalization of KODIAK. Tractable reasoning procedures for $\mathcal{RDL}$ are briefly revealed in section 4. In the last part of the paper, first a solution to the *Reified Relation* and *QUA-link* problems in the context of Kl-One-based description logics is presented. Then, we show how this solution can be used to map (a subset) of $\mathcal{RDL}$ in Kl-One.

[1]This work started from a discussion with Bernhard Nebel.

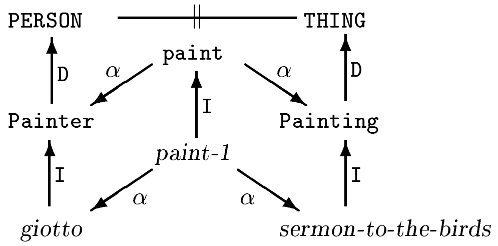

Figure 1: A sample KODIAK knowledge base.

2 KODIAK

The observations of the previous section lead us to consider a formalization of KODIAK, to be eventually compared with the expressivity of a Kl-One-based description logic. KODIAK is a description language not belonging to the Kl-One family. It is an attempt to reconstruct a description language from a different point of view.

KODIAK is a relation-based representation language, as opposed to Kl-One, which is a concept-based representation language. We consider here only a simplified version of the language. The syntactic types of the language are *Relations*, *Aspectuals*, *Absolutes*; the basic operators of the language are *Manifest*, *Dominate*, *Instantiate*, *Disjointness*. Figure 1 shows an example. The relation `paint` manifests the two aspectuals `Painter` and `Painting` which are dominated by the mutually disjoint absolutes `PERSON` and `THING`; *paint-1* is a relation instantiating `paint` and manifesting the instances *giotto* and *sermon-to-the-birds*. It is important to notice that this schema defines completely all the mentioned entities. The aspectual `Painter` *is defined* as being the first participant of the relation `paint`, whereas the relation `Paint` is defined as having two participants, namely `Painter` and `Painting`. On the other hand, the absolute `PERSON` is defined only as being disjoint from the absolute `THING`.

3 The Formal Language $\mathcal{RDL}$

We introduce in this section a very preliminary formalization of KODIAK, called $\mathcal{RDL}$. A $\mathcal{RDL}$ knowledge base is a collection of statements of the following form:

DOMINATE $(\mathtt{C}_1, \mathtt{C}_2)$	$\mathtt{C}_1, \mathtt{C}_2$ with same arity
MANIFEST $(\mathtt{R}, \mathtt{C}_1 \ldots \mathtt{C}_n)$	$n \geq 1$
DISJOINT $(\mathtt{C}_1 \ldots \mathtt{C}_m)$	$m \geq 2$; $\mathtt{C}_i$ with same arity
$\mathtt{C}$ $(a_1 \ldots a_l)$	$l \geq 1$

where '$\mathtt{C}$' is a syntactic object of the sort *concept*, '$\mathtt{R}$' is a syntactic object of the sort *relation* and 'a' is a syntactic object of sort *individual*; the sort *relation* is a sub-sort of the sort *concept*, i.e. any relation is a concept. A relation is associated to its arity, which is defined by the number of the other concepts appearing in the MANIFEST statement in which they appear as the first argument. Non relational concepts can be also seen as relations of zero arity.

To give a semantics to $\mathcal{RDL}$, let us associate to each relation of arity n a predicate symbol of arity $n+1$, and to each individual a constant symbol in a classical logic. For sake of simplicity, syntactic objects in $\mathcal{RDL}$ and the corresponding semantic symbols in the logic will have the same representation, being clear from the context their meaning. Relations should satisfy the following axioms:

$$\forall R. \forall i. \exists \bar{x}. R(i, \bar{x})$$
$$\forall R_1, R_2. \forall i. \forall \bar{x}, \bar{y}. (R_1(i, \bar{x}) \wedge R_2(i, \bar{y})) \rightarrow (\bar{x} = \bar{y})$$

These axioms state a kind of correspondence between the first argument of the predicate and the tuple formed by the rest of the arguments. The first argument – called *index* – represents in some sense the particular instance of the relation, whereas the rest of the arguments represent the instances of the fillers of the relation. Given an index for a relation, the corresponding fillers of the relation are uniquely determined; however, a tuple of fillers of a relation may correspond to more than one index. Our knowledge base is so rewritten in $\mathcal{RDL}$:

DOMINATE (`PERSON`, `Painter`)
DOMINATE (`THING`, `Painting`)
DISJOINT (`PERSON`, `THING`)
MANIFEST (`paint`, `Painter`, `Painting`)
`paint` (*paint-1*, *giotto*, *sermon-to-the-birds*)
`Painter` (*giotto*)
`Painting` (*sermon-to-the-birds*).

Please note that the last two statements are redundant, i.e. they can be deduced from the former statements.

The semantics of the operators is given as a mapping in the classical logic and it as follows[2]:

- DOMINATE $(\mathtt{C_1}, \mathtt{C_2})$ iff $\forall i, \bar{x}. C_2(i, \bar{x}) \rightarrow C_1(i, \bar{x})$
- MANIFEST $(\mathtt{R}, \mathtt{C_1}, \ldots, \mathtt{C_n})$ iff
  $\forall x_1 \ldots x_n \exists i. R(i, x_1 \ldots x_n) \rightarrow C_1(x_1) \wedge \ldots \wedge C_n(x_n)$
  $\forall x_1. C_1(x_1) \rightarrow \exists i, x_2 \ldots x_n. R(i, x_1 \ldots x_n)$
  $\vdots$
  $\forall x_n. C_n(x_n) \rightarrow \exists i, x_1 \ldots x_{n-1}. R(i, x_1 \ldots x_n)$
- DISJOINT $(\mathtt{C_1} \ldots \mathtt{C_n})$ iff
  DISJOINT $(\mathtt{C_i}, \mathtt{C_j})$ for all $i, j = 1 \ldots n$ and $i \neq j$
- DISJOINT $(\mathtt{C_1}, \mathtt{C_2})$ iff $\forall i, \bar{x}. \neg(C_1(i, \bar{x}) \wedge C_2(i, \bar{x}))$
- $\mathtt{C}$ $(a_1 \ldots a_l)$ iff $C(a_1 \ldots a_l)$

In $\mathcal{RDL}$ relations can be the arguments of other relations; for example, the following is a knowledge base expressing "Vania is watching at Giotto painting the *Sermon to the birds*":

MANIFEST (`see`, `Experiencer`, `experience`)
DOMINATE (`experience`, `paint`)
`see` (*see-1*, *vania*, *paint-1*)
`paint` (*paint-1*, *giotto*, *sermon-to-the-birds*)

4 Reasoning in $\mathcal{RDL}$

We have defined in $\mathcal{RDL}$ as usual three reasoning services: satisfiability, subsumption between concepts of the same arity and instance checking. We have found sound and complete algorithms for these services, running in polynomial time [Rabito,1994]. Figure 2 (a) shows a sound and complete set of inference rules for the derivation of disjointness and subsumption. This set of rules is inspired by the work of [Atzeni and Parker Jr.,1986]. The inference rules presented in the whole figure 2 (a) + (b) are a sound and complete set of inference rules also for the derivation of ground instances.

A $\mathcal{RDL}$ knowledge base Σ is satisfiable (or consistent) if and only if none of the following conditions holds:

- $\Sigma \vdash$ DISJOINT(C, C)
- $\Sigma \vdash C(i, a_1, \ldots, a_n),\ D(i, a_1, \ldots, a_n),$ DISJOINT(C, D)
- $\Sigma \vdash$ DISJOINT(C_i, D_j), MANIFEST$(R, C_1 \ldots C_n)$, MANIFEST$(S, D_1 \ldots D_m)$, $R(i_1, a_1, \ldots, a_n),\ S(i_2, b_1, \ldots, b_m),\ a_i = b_j$ for some $i, j,\ 1 \leq i \leq n,\ 1 \leq j \leq m$

A concept C subsumes a concept D with respect to a $\mathcal{RDL}$ knowledge base if and only if a statement of the form $C \sqsupseteq D$ can be derived by applying the rules to the knowledge base augmented with the statements defining C and D.

Given a $\mathcal{RDL}$ knowledge base Σ, an ordered set of individuals $i, a_1, \ldots, a_n$ and a concept C, the *instance checking problem* is to test whether $\Sigma \models C(i, a_1, \ldots, a_n)$. Informally, it is the inference task checking whether the individual i is an instance of the concept C and, if C is a relational concept, whether $a_1, \ldots, a_n$ are instances of the fillers of the relation itself. $\Sigma \models C(i, a_1, \ldots, a_n)$

[2]In the following, for sake of simplicity it is intended that in the case of non relational concepts, all the arguments of the corresponding predicate – but the first – should be omitted, i.e. $C(i, a_1, \ldots, a_n)$ becomes $C(i)$.

(a)

- $\vdash C \sqsupseteq C$
- $\textsc{dominate}(C_1, C_2) \vdash C_1 \sqsupseteq C_2$
- $C_1 \sqsupseteq C_2,\ C_2 \sqsupseteq C_3 \vdash C_1 \sqsupseteq C_3$
- $R \sqsupseteq S,\ \textsc{manifest}(R, C_1 \dots C_n),$ $\textsc{manifest}(S, D_1 \dots D_n) \vdash C_i \sqsupseteq D_i$ for all $i = 1, \dots, n$
- $\textsc{disjoint}(C_1 \dots C_n) \vdash \textsc{disjoint}(C_i, C_j)$ for all $i, j = 1, \dots, n$ and $i \neq j$
- $\textsc{disjoint}(C_1, C_1) \vdash \textsc{disjoint}(C_1, C_2)$
- $\textsc{disjoint}(C_1, C_2),\ C_2 \sqsupseteq C_3 \vdash \textsc{disjoint}(C_1, C_3)$
- $\textsc{disjoint}(C_1, C_1) \vdash C_2 \sqsupseteq C_1$
- $\textsc{manifest}(R, C_1 \dots C_n),\ \textsc{manifest}(S, D_1 \dots D_n),$ $\textsc{disjoint}(C_i, D_i) \vdash \textsc{disjoint}(R, S)$ for some $i, 1 \leq i \leq n$
- $\textsc{manifest}(R, C_1 \dots C_n),\ \textsc{disjoint}(R, R) \vdash$ $\textsc{disjoint}(C_i, C_i)$ for all $i = 1, \dots, n$

(b)

- $C_1(i, a_1, \dots, a_n),\ C_2 \sqsupseteq C_1 \vdash C_2(i, a_1, \dots, a_n)$
- $\textsc{manifest}(R, C_1 \dots C_{j-1},\ C\ , C_{j+1} \dots C_n),$ $R(j, b_1 \dots b_{j-1},\ i\ , b_{j+1} \dots b_n),\ D(i, a_1 \dots a_n) \vdash$ $C(i, a_1, \dots, a_n)$

Figure 2: Inference rules for the derivation of disjointness, subsumption and ground instances.

holds if and only if $C(i, a_1, \dots, a_n)$ can be derived by applying the rules.

In [Rabito,1994] an extension of the language with other operators (e.g. equality, disjointness in context) and services (e.g. *inheritance* and subsumption between concepts with different arity, concretion [Jacobs,1992; Bayer and Vilain,1991; Vilain,1993]) is presented.

5 Reified Relations and QUA-links in KL-ONE

In this section we introduce the *Reified Relation* and *QUA-link* problems in the context of KL-ONE-based description logics.

Reification is an operator which allows for both relation-as-role and relation-as-concept interpretations of a binary relation. Formal semantics for reification can be given by the following axiom:

$$\forall x, y.\, R(x, y) \Leftrightarrow \exists z.\, \hat{R}(z) \wedge \Delta(z) = x \wedge \Theta(z) = y$$

where $\hat{R}$ is the reified relation of R, Δ is a function mapping to the domain of the relation and Θ is a function mapping to the range. This operation is crucial for NL applications, e.g. to represent that a noun phrase modified by a prepositional phrase can have the same meaning of the NP modified by a relative phrase. This is the case

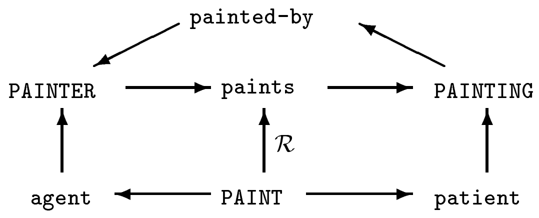

Figure 3: Reification in KL-ONE

if the interpretation of the preposition is the same as the one of the relative verb. It is assumed, as it is usual, that prepositions are mapped to roles and verbs to concepts. For example the following sentences should have the same meaning:

Show me a fresco of *Giotto's.*

Show me a fresco painted by *Giotto.*

In this example – see figure 3 – the concept `PAINT` associated to the verb "paint" reifies the role `paints` associated to the preposition "of", the Δ functional role corresponds to the `agent` role of the verb, and the Θ functional role corresponds to the `patient` role of the verb.

Informal or incomplete solutions to the Reified Relation problem have been presented in [Bateman *et al.*,1990; Kobsa,1991; MacGregor,1993].

At the 1981 KL-ONE workshop [Schmolze and Brachman,1982] the QUA-link was informally introduced. The QUA-link was defined as

> a kind of inheritance cable pointing from the concept being defined to the *RoleSet* of another concept taken as its definitional context. From a purely syntactic point of view, the QUA-defined concept was considered *subC* to the *V/R* of the *RoleSet* it pointed to.

Formal semantics capturing the idea of the QUA-link is given by the following axiom[3]:

$$(\bowtie C\, R.\, D)^{\mathcal{I}} = \{x \mid x \in D^{\mathcal{I}} \wedge \exists y.\, y \in C^{\mathcal{I}} \wedge \langle y, x \rangle \in R^{\mathcal{I}}\}$$

As an example, consider how to represent the `PAINT` relation, using the QUA operator:

$$\begin{array}{lcl} \texttt{PAINT} & \dot{\sqsubseteq} & \exists\texttt{agent} \sqcap \exists\texttt{patient} \sqcap \\ & & \leq\!1\,\texttt{agent} \sqcap \leq\!1\,\texttt{patient} \\ \texttt{PAINTER} & \doteq & \bowtie \texttt{PAINT}\,\texttt{agent.PERSON} \\ \texttt{PAINTING} & \doteq & \bowtie \texttt{PAINT}\,\texttt{patient.THING} \end{array}$$

The latter definitions are equivalent to the following definitions:

$$\begin{array}{lcl} \texttt{PAINTER} & \doteq & \texttt{PERSON} \sqcap \exists\texttt{agent}^{-1}.\texttt{PAINT} \\ \texttt{PAINTING} & \doteq & \texttt{THING} \sqcap \exists\texttt{patient}^{-1}.\texttt{PAINT} \end{array}$$

[3] This was discovered after a discussion with Fabrizio Sebastiani.

It is worth noting that the latter definitions do not involve terminological cycles, as one would expect[4]. In fact, a naive translation would include cycles, since `PAINTER` is being defined as being the `agent` of `PAINT` and, at the same time, `PAINT` is being defined as having a `PAINTER` as `agent`. So, we do not need the full expressive power of Converse Deterministic PDL, as it is shown in [De Giacomo and Lenzerini,1994] in order to represent correctly n-ary relations in a description logic[5]. In fact, this definition can be expressed in the polynomial description logic of [Buchheit *et al.*,1994a; 1994b].

In this context, the reified relation problem is exactly solved by means of the following equations:

$$\begin{array}{lcl}
\hat{\mathtt{R}} & \dot{\sqsubseteq} & \exists\Delta \sqcap \exists\Theta \sqcap \leq 1\,\Delta \sqcap \leq 1\,\Theta \\
\mathtt{C}_{\Delta} & \doteq & \exists\Delta^{-1}.\hat{\mathtt{R}} \\
\mathtt{C}_{\Theta} & \doteq & \exists\Theta^{-1}.\hat{\mathtt{R}} \\
\mathtt{R} & \doteq & (\Delta^{-1}\,|\hat{\mathtt{R}}) \circ \Theta
\end{array}$$

These equations can be generalized to the n-ary case:

$$\begin{array}{lcl}
\hat{\mathtt{R}} & \dot{\sqsubseteq} & \exists\mathtt{a}_1 \sqcap \ldots \sqcap \exists\mathtt{a}_n \sqcap \leq 1\,\mathtt{a}_1 \sqcap \ldots \sqcap \leq 1\,\mathtt{a}_n \\
\mathtt{C}_{\mathtt{i}} & \doteq & \exists{\mathtt{a}_{\mathtt{i}}}^{-1}.\hat{\mathtt{R}} \\
\mathtt{R}_{\mathtt{ij}} & \doteq & ({\mathtt{a}_{\mathtt{i}}}^{-1}\,|\hat{\mathtt{R}}) \circ \mathtt{a}_{\mathtt{j}}
\end{array}$$

It can be shown that $\mathcal{RDL}$ knowledge bases without disjointness can be mapped in KL-ONE knowledge bases using the equations cited above. However, even if this mapping maintains the tractability of the reasoning procedures, it does not consider disjointness, whose complexity is still polynomial in $\mathcal{RDL}$, but it is not yet known in KL-ONE.

The comparison between $\mathcal{RDL}$ and KL-ONE should also take into account the *Entity-Relationship* (ER) semantic data models framework [Calvanese *et al.*,1994] and the Propositional Knowledge Representation framework [D'Aloisi and Castelfranchi,1993].

References

[Atzeni and Parker Jr., 1986] Paolo Atzeni and D. Stott Parker Jr. Formal properties of net-based knowledge representation schemes. In *Proc. of the 2nd Conference on Data Engineering*, Los Angeles, CA, February 1986. A long version appears as Technical Report CNR-IASI 135, Rome, Italy.

[Bateman *et al.*, 1990] John A. Bateman, Robert T. Kasper, Johanna D. Moore, and Richard A. Whitney. A general organization of knowledge for natural language processing: the PENMAN upper model. Technical report, USC/Information Science Institute, Marina del Rey CA, March 1990.

[4]This was suggested by Werner Nutt.

[5]The goal of [De Giacomo and Lenzerini,1994] is to model n-ary relations: a n-ary relation denotes a set of tuples, whereas a relation defined with a QUA-link denotes a multiset of tuples. In the latter case, we speak of *instances* of relations.

[Bayer and Vilain, 1991] Samuel Bayer and Marc Vilain. The relation-based knowledge representation of King-Kong. In *Working Notes of the AAAI Spring Symposium "Implemented Knowledge Representation and Reasoning Systems"*, pages 31–44, March 1991.

[Buchheit *et al.*, 1994a] M. Buchheit, F. M. Donini, W. Nutt, and A. Schaerf. Refining the structure of terminological systems: Terminology = schema + views. In *Proc. of AAAI-94*, 1994. To appear.

[Buchheit *et al.*, 1994b] Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, and Martin Staudt. Subsumption between queries to object-oriented databases. *Information Systems*, 19(1):33–54, 1994.

[Calvanese *et al.*, 1994] Diego Calvanese, Maurizio Lenzerini, and Daniele Nardi. A unified framework for class-based representation formalisms. In *Proc. of KR-94*, Bonn D, May 1994.

[D'Aloisi and Castelfranchi, 1993] Daniela D'Aloisi and Cristiano Castelfranchi. Propositional and terminological knowledge representation. *JETAI*, 5(2-3):149–166, 1993.

[De Giacomo and Lenzerini, 1994] Giuseppe De Giacomo and Maurizio Lenzerini. Description logics with inverse roles, functional restrictions, and n-ary relations. In *Proc. of the 4th European Workshop on Logics in AI, JELIA-94*, September 1994.

[Jacobs, 1992] Paul S. Jacobs. TRUMP: A transportable language understanding program. *International Journal of Intelligent Systems*, 7:245–276, March 1992.

[Kobsa, 1991] Alfred Kobsa. Reification in SB-ONE. In *International Workshop on Terminological Logics*, DFKI Document D-91-13, pages 72–74, May 1991.

[MacGregor, 1993] Robert M. MacGregor. Representing reified relations in LOOM. *Journal of Experimental and Theoretical Artificial Intelligence*, 5:179–183, 1993.

[Norvig, 1987] Peter Norvig. Unified theory of inference for text understanding. UCB/CSD Report 87/339, University of California, Berkeley CA, January 1987.

[Rabito, 1994] Vania Rabito. Studio di un linguaggio per la rappresentazione della conoscenza basato su relazioni. Master's thesis, University of Milan, Italy, July 1994. In italian.

[Schmolze and Brachman, 1982] James G. Schmolze and Ronald J. Brachman, editors. *Proceedings of the 1981 KL-ONE Workshop*, Fairchild Technical Report No. 618, May 1982.

[Vilain, 1993] Marc Vilain. Validation of terminological inference in an information extraction task. In *Proc. of the 1993 ARPA Human Language Workshop*. Morgan Kaufman, 1993.

[Wilensky, 1987] Robert Wilensky. Some problems and proposals for knowledge representation. UCB/CSD Report 87/351, University of California, Berkeley, CA, May 1987.