Ontology-based reasoning techniques for multimedia interpretation and retrieval

Ralf Möller[1] and Bernd Neumann[2]

[1] Hamburg University of Technology, r.f.moeller@tu-harburg.de
[2] University of Hamburg, neumann@informatik.uni-hamburg.de

1 Introduction

In this chapter we show how formal knowledge representation and reasoning techniques can be used for the retrieval and interpretation of multimedia data. This section explains what we mean by an "interpretation" using examples of audio and video interpretation. Intuitively, interpretations are descriptions of media data at a high abstraction level, exposing interrelations and coherencies. In Section 2.3, we introduce description logics (DLs) as the formal basis for ontology languages of the OWL (Web Ontology Language) family and for the interpretation framework described in subsequent sections. As a concrete example, we consider the interpretation of images describing a sports event in Section 3. It is shown that interpretations can be obtained by abductive reasoning, a general interpretation framework is presented. Stepwise construction of an interpretation can be viewed as navigation in the compositional and taxonomical hierachies spanned by a conceptual knowledge base.

What do we mean by "interpretation" of media objects? Consider the image shown in Figure 3.1. One can think of the image as a set of primitive objects such as persons, garbage containers, a garbage truck, a bicycle, traffic signs, trees etc. An interpretation of the image is a description which "makes sense" of these primitive objects. In our example, the interpretation could include the assertions "two workers empty garbage containers into a garbage truck" and "a mailman distributes mail" expressed in some knowledge representation language.

When including the figure caption into the interpretation process, we have a multimodal interpretation task which in this case involves visual and textual media objects. The result could be a refinement of the assertions above in terms of the location "in Hamburg". Note that the interpretation describes activities extending in time although it is only based on a snapshot. Interpretations may generally include hypotheses about things outside the temporal and spatial scope of the available media data.

An interpretation is a "high-level" description of media data in the sense that it involves terms which abstract from details at lower representation levels. This is typ-

Fig. 3.1. Street scene in Hamburg.

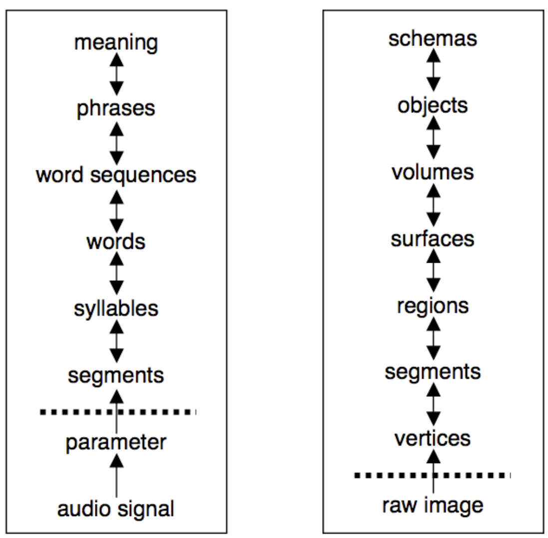

Fig. 3.2. Typical level structure of media interpretation systems exemplified by HEARSAY II and VISIONS. Signal processing (below dotted line) transforms the raw input signals into primitive media objects, various interpretation processes lead up to higher-level interpretations.

ical for meaningful descriptions in human language and hence also a desirable goal for machine interpretation. Media interpretation is therefore often structured as a process computing higher-level representations from lower-level ones. Figure 3.2 shows the level structure of two early interpretation systems, the speech recognition system

HEARSAY-II (Erman, Hayes-Roth, Lesser and Reddy 1980) and the image interpretation system VISIONS (Hanson and Riseman 1978).

The basic structure exemplified by each of the two systems also applies to interpretation systems in general: signal processing procedures first transform raw media data into primitive media objects by low-level processing steps. Then higher-level descriptions are determined based on the primitive media objects. The low-level processing steps are often called "analysis" (e.g. image analysis, speech analysis), the high-level steps constitute the interpretation process.

It is useful to view interpretation as a process which is both based on the general conceptual knowledge and the concrete contextual knowledge which an agent may possess. The term "contextual knowledge" covers specific prior knowledge relevant for the interpretation which the agent may possess (e.g. spatial and temporal context of a video clip) as well as the knowledge about the current task of the agent (e.g. recognizing criminal acts vs. recognizing sports events). The knowledge-based structure of an image sequence interpretation system is shown in Figure 3.3.

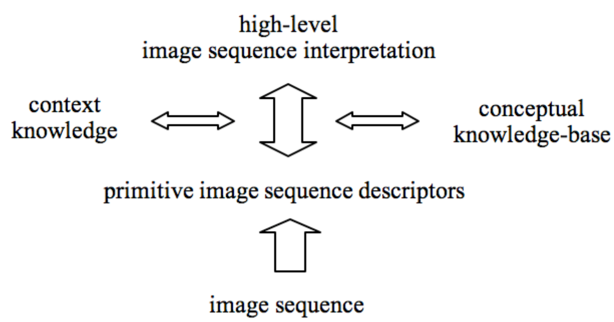

Fig. 3.3. Knowledge-based structure of a system for image sequence interpretation.

The concepts represented in the conceptual knowledge base typically describe configurations of lower-level entities forming some interesting higher-level entity, for example a configuration of an athlete and a horizontal bar forming a "high jump" event. We call such concepts "aggregates" as they combine several components to a larger whole. Aggregates form a compositional hierarchy, in addition to the taxonomical hierarchy induced by logic-based concept definitions. In a description logic setting, an aggregate has the generic structure shown in Figure 3.4 (Neumann and Möller 2006). An aggregate is defined by (1) inheritance from parent concepts, (2) roles relating the aggregate to parts, and (3) constraints relating parts to each other. Instantiations of aggregates are at the core of media interpretations.

In summary, interpretations have the following characteristics: they

- involve several objects;
- depend on the temporal or spatial relations between parts;
- describe the data in qualitative terms, omitting detail;

$$
\begin{aligned}
Aggregate\_Concept \equiv\ & Parent\_Concept_1 \sqcap \ldots \sqcap Parent\_Concept_n \sqcap \\
& \exists_{\geq m_1} hasPartRole.Part\_Concept_1 \sqcap \\
& \ldots \\
& \exists_{\geq m_k} hasPartRole.Part\_Concept_k \sqcap \\
& constraints\ between\ parts
\end{aligned}
$$

Fig. 3.4. Generic structure of a definition for an aggregate concept (ontology design pattern).

- exploit contextual information;
- include inferred facts, not explicit in the data;
- are based on conceptual knowledge about the application domain.

The chapter is structured as follows. We first describe how ontology-based information retrieval can be formalized using description-logic inference problems (Section 2). Introducing the necessary technical background we demonstrate for what purpose the output of media interpretation can be used, and, thereby, derive requirements for the media interpretation process. Then, in Section 3, the automatic construction of media interpretations is investigated. Techniques for dealing with uncertain and ambiguous interpretations are presented in Section 4. We conclude in Section 5.

In summary it is the purpose of this chapter to show that interpretations can be computed in a formal knowledge representation framework using various reasoning processes. This has multiple potential benefits. First, the complex computational process media interpretation is realised via standardised reasoning procedures, i.e. by programs which have been conceptually shown to meet correctness and completeness conditions, and have been implemented as reusable tools for a wide range of applications. Second, the terms by which interpretations are expressed are embedded in a sharable ontology which provides a transparent declarative representation with well-defined semantics. Furthermore, ontologies constitute resources not only for media interpretation but also for other tasks dealing with semantic content, such as information retrieval, communication, documentation, and various engineering processes.

2 Ontology-based information retrieval

A task addressed in this chapter is information retrieval from the Semantic Web. Media data with semantic annotations will be an important part of the information provided by the Semantic Web. It is well-known that the Semantic Web representation language OWL can be formally described using description logics and that reasoning services of description logics apply to the Semantic Web. One of those services is media interpretation, which is the main topic in this chapter. In the context of the Semantic Web, media interpretation may provide the bridge from low-level media annotation to information retrieval in higher-level terms. One could think, for example, of an off-line service enriching low-level media annotations with high-level interpretations. Alternatively, media interpretation could be part of a retrieval service, providing interpretations on-the-fly. Information retrieval w.r.t. high-level interpretations is different from so-called content-based retrieval (e.g. retrieval based on similarity measures w.r.t.

colour histograms, strings, or other low-level features). In our view, high-level interpretations are attached to media objects as metadata, which are specified using ontology languages.

2.1 Ontology languages based on description logics

Ontology languages of the OWL family, which provide the skeleton for research on information retrieval based on high-level media interpretations, are based on description logics. In this section we introduce the logical basis of several ontology languages of the OWL family, define their semantics, and specify corresponding reasoning services. In the following subsections, we start with so-called expressive description logics (approximately corresponding to, but slightly more expressive than OWL Lite), introduce additional constructs afterwards (corresponding to OWL DL, OWL 1.1), and also specify other fragments of first-order logic, some of which have also been standardised by activities of the World-Wide Web Consortium (W3C). For more details see (Baader, Calvanese, McGuinness, Nardi and Patel-Schneider 2003).

Expressive Description Logics: Syntax and Semantics

The DL $\mathcal{ALCQHI}_{R^+}(\mathcal{D})^-$ which is also known as $\mathcal{SHIQ}$ is briefly introduced as follows. We assume five disjoint sets: a set of concept names C, a set of role names R, a set of feature names F, a set of individual names O and a set of names for (concrete) objects O_C. The mutually disjoint subsets P and T of R denote non-transitive and transitive roles, respectively ($R = P \cup T$). $\mathcal{ALCQHI}_{R^+}(\mathcal{D})^-$ is introduced in Figure 3.5 using a standard Tarski-style semantics with an interpretation $\mathcal{I}_\mathcal{D} = (\Delta^\mathcal{I}, \Delta^\mathcal{D}, \cdot^\mathcal{I})$ where $\Delta^\mathcal{I} \cap \Delta^\mathcal{D} = \emptyset$ holds. A variable assignment α maps concrete objects to values in $\Delta^\mathcal{D}$.

In accordance with (Baader and Hanschke 1991) we also define the notion of a concrete domain. A *concrete domain* $\mathcal{D}$ is a pair $(\Delta_\mathcal{D}, \Phi_\mathcal{D})$, where $\Delta_\mathcal{D}$ is a set called the domain, and $\Phi_\mathcal{D}$ is a set of predicate names. The interpretation function maps each predicate name P from $\Phi_\mathcal{D}$ with arity n to a subset $\mathsf{P}^\mathcal{I}$ of $\Delta_\mathcal{D}^n$. Concrete objects from O_C are mapped to an element of $\Delta^\mathcal{D}$. We assume that $\bot_\mathcal{D}$ is the negation of the predicate $\top_\mathcal{D}$. A concrete domain $\mathcal{D}$ is called *admissible* iff the set of predicate names $\Phi_\mathcal{D}$ is closed under negation and $\Phi_\mathcal{D}$ contains a name $\top_\mathcal{D}$ for $\Delta_\mathcal{D}$, and the satisfiability problem $\mathsf{P}_1^{\mathsf{n}_1}(\mathsf{x}_{11}, \ldots, \mathsf{x}_{1\mathsf{n}_1}) \wedge \ldots \wedge \mathsf{P}_\mathsf{m}^{\mathsf{n}_\mathsf{m}}(\mathsf{x}_{\mathsf{m}1}, \ldots, \mathsf{x}_{\mathsf{m}\mathsf{n}_\mathsf{m}})$ is decidable (m is finite, $\mathsf{P}_\mathsf{i}^{\mathsf{n}_\mathsf{i}} \in \Phi_\mathcal{D}$, n_i is the arity of P_i, and x_jk is a concrete object).

If $\mathsf{R}, \mathsf{S} \in R$ are role names, then $\mathsf{R} \sqsubseteq \mathsf{S}$ is called a *role inclusion axiom*. A *role hierarchy* $\mathcal{R}$ is a finite set of role inclusion axioms. Then, we define $\sqsubseteq^*$ as the reflexive transitive closure of $\sqsubseteq$ over such a role hierarchy $\mathcal{R}$. Given $\sqsubseteq^*$, the set of roles $\mathsf{R}^\downarrow = \{\mathsf{S} \in R \mid \mathsf{S} \sqsubseteq^* \mathsf{R}\}$ defines the *sub-roles* of a role R. R is called a super-role of S if $\mathsf{S} \in \mathsf{R}^\downarrow$. We also define the set $S := \{\mathsf{R} \in P \mid \mathsf{R}^\downarrow \cap T = \emptyset\}$ of *simple* roles that are neither transitive nor have a transitive role as sub-role. Due to undecidability issues, number restrictions are only allowed for simple roles (Horrocks, Sattler and Tobies 2000). In concepts, inverse roles R^{-1} (or S^{-1}) may be used instead of role names R (or S). If C and D are concepts, then $\mathsf{C} \sqsubseteq \mathsf{D}$ is a terminological axiom (*generalized concept*

(a)

Syntax	Semantics
Concepts ($\mathsf{R} \in R$, $\mathsf{S} \in S$, and $\mathsf{f}, \mathsf{f_i} \in F$)	
A	$\mathsf{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ (A *is a concept name*)
$\neg \mathsf{C}$	$\Delta^{\mathcal{I}} \setminus \mathsf{C}^{\mathcal{I}}$
$\mathsf{C} \sqcap \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \cap \mathsf{D}^{\mathcal{I}}$
$\mathsf{C} \sqcup \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \cup \mathsf{D}^{\mathcal{I}}$
$\exists \mathsf{R} \,.\, \mathsf{C}$	$\{\mathsf{a} \in \Delta^{\mathcal{I}} \mid \exists \mathsf{b} \in \Delta^{\mathcal{I}} : (\mathsf{a}, \mathsf{b}) \in \mathsf{R}^{\mathcal{I}} \wedge \mathsf{b} \in \mathsf{C}^{\mathcal{I}}\}$
$\forall \mathsf{R} \,.\, \mathsf{C}$	$\{\mathsf{a} \in \Delta^{\mathcal{I}} \mid \forall \mathsf{b} \in \Delta^{\mathcal{I}} : (\mathsf{a}, \mathsf{b}) \in \mathsf{R}^{\mathcal{I}} \Rightarrow \mathsf{b} \in \mathsf{C}^{\mathcal{I}}\}$
$\exists_{\geq n} \mathsf{S} \,.\, \mathsf{C}$	$\{\mathsf{a} \in \Delta^{\mathcal{I}} \mid \Vert\{\mathsf{x} \mid (\mathsf{a}, \mathsf{x}) \in \mathsf{S}^{\mathcal{I}}, \mathsf{x} \in \mathsf{C}^{\mathcal{I}}\}\Vert \geq \mathsf{n}\}$
$\exists_{\leq m} \mathsf{S} \,.\, \mathsf{C}$	$\{\mathsf{a} \in \Delta^{\mathcal{I}} \mid \Vert\{\mathsf{x} \mid (\mathsf{a}, \mathsf{x}) \in \mathsf{S}^{\mathcal{I}}, \mathsf{x} \in \mathsf{C}^{\mathcal{I}}\}\Vert \leq \mathsf{m}\}$
$\exists \mathsf{f}_1, \ldots, \mathsf{f_n} \,.\, \mathsf{P}$	$\{\mathsf{a} \in \Delta^{\mathcal{I}} \mid \exists \mathsf{x}_1, \ldots, \mathsf{x_n} \in \Delta^{\mathcal{D}} : (\mathsf{a}, \mathsf{x}_1) \in \mathsf{f}_1{}^{\mathcal{I}} \wedge \ldots \wedge (\mathsf{a}, \mathsf{x_n}) \in \mathsf{f_n}{}^{\mathcal{I}} \wedge$ $(\mathsf{x}_1, \ldots, \mathsf{x_n}) \in \mathsf{P}^{\mathcal{I}}\}$
$\forall \mathsf{f}_1, \ldots, \mathsf{f_n} \,.\, \mathsf{P}$	$\{\mathsf{a} \in \Delta^{\mathcal{I}} \mid \forall \mathsf{x}_1, \ldots, \mathsf{x_n} \in \Delta^{\mathcal{D}} : (\mathsf{a}, \mathsf{x}_1) \in \mathsf{f}_1{}^{\mathcal{I}} \wedge \ldots \wedge (\mathsf{a}, \mathsf{x_n}) \in \mathsf{f_n}{}^{\mathcal{I}} \Rightarrow$ $(\mathsf{x}_1, \ldots, \mathsf{x_n}) \in \mathsf{P}^{\mathcal{I}}\}$
Roles and Features	
R	$\mathsf{R}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
f	$\mathsf{f}^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow \Delta^{\mathcal{D}}$ (features are partial functions)

$\Vert \cdot \Vert$ denotes the cardinality of a set, and $n, m \in \mathbb{N}$ with $n > 1,\ m > 0$.

(b)

Axioms	
Syntax	Satisfied if
$\mathsf{R} \in T$	$\mathsf{R}^{\mathcal{I}} = (\mathsf{R}^{\mathcal{I}})^{+}$
$\mathsf{R} \sqsubseteq \mathsf{S}$	$\mathsf{R}^{\mathcal{I}} \subseteq \mathsf{S}^{\mathcal{I}}$
$\mathsf{C} \sqsubseteq \mathsf{D}$	$\mathsf{C}^{\mathcal{I}} \subseteq \mathsf{D}^{\mathcal{I}}$

(c)

Assertions ($\mathsf{a}, \mathsf{b} \in O$, $\mathsf{x}, \mathsf{x_i} \in O_C$)	
Syntax	Satisfied if
$\mathsf{a} : \mathsf{C}$	$\mathsf{a}^{\mathcal{I}} \in \mathsf{C}^{\mathcal{I}}$
$(\mathsf{a}, \mathsf{b}) : \mathsf{R}$	$(\mathsf{a}^{\mathcal{I}}, \mathsf{b}^{\mathcal{I}}) \in \mathsf{R}^{\mathcal{I}}$
$(\mathsf{a}, \mathsf{x}) : \mathsf{f}$	$(\mathsf{a}^{\mathcal{I}}, \alpha(\mathsf{x})) \in \mathsf{f}^{\mathcal{I}}$
$(\mathsf{x}_1, \ldots, \mathsf{x_n}) : \mathsf{P}$	$(\alpha(\mathsf{x}_1), \ldots, \alpha(\mathsf{x_n})) \in \mathsf{P}^{\mathcal{I}}$
$\mathsf{a} = \mathsf{b}$	$\mathsf{a}^{\mathcal{I}} = \mathsf{b}^{\mathcal{I}}$
$\mathsf{a} \neq \mathsf{b}$	$\mathsf{a}^{\mathcal{I}} \neq \mathsf{b}^{\mathcal{I}}$

Fig. 3.5. Syntax and Semantics of $\mathcal{ALCQHI}_{R^+}(\mathcal{D})^-$.

inclusion or *GCI*). $\mathsf{C} \equiv \mathsf{D}$ is used as an abbreviation for two GCIs $\mathsf{C} \sqsubseteq \mathsf{D}$ and $\mathsf{D} \sqsubseteq \mathsf{C}$. A finite set of terminological axioms $\mathcal{T}_{\mathcal{R}}$ is called a *terminology* or *TBox* w.r.t. to a given role hierarchy $\mathcal{R}$. The reference to $\mathcal{R}$ is omitted in the following if we use $\mathcal{T}$. An *ABox* $\mathcal{A}$ is a finite set of assertional axioms as defined in Figure 3.5c.

An interpretation $\mathcal{I}$ is a *model* of a concept C (or *satisfies* a concept C) iff $\mathsf{C}^{\mathcal{I}} \neq \emptyset$ and for all $\mathsf{R} \in R$ it holds that iff $(x, y) \in \mathsf{R}^{\mathcal{I}}$ then $(y, x) \in (\mathsf{R}^{-1})^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ iff it satisfies all axioms in $\mathcal{T}$ (see Figure 3.5b). An interpretation $\mathcal{I}$ is a model of an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ iff it is a model of $\mathcal{T}$ and satisfies all assertions in $\mathcal{A}$ (see Figure 3.5c). Different individuals are mapped to different domain objects (unique name assumption).

Reasoning about objects from other domains (so-called concrete domains, e.g. for real numbers) is very important for practical applications, in particular, in the context of the Semantic Web. For instance, one might want to express intervals for integer values ("the price range is between 200 and 300 Euro"), state the relationship between the Fahrenheit and Celsius scales, or describe linear inequalities ("the total price for the three goods must be below 60 Euro"). In (Baader and Hanschke 1991) the description logic $\mathcal{ALC}(\mathcal{D})$ is investigated and it is shown that, provided a decision procedure for the concrete domain $\mathcal{D}$ exists, the logic $\mathcal{ALC}(\mathcal{D})$ is decidable. Unfortunately, adding concrete domains to expressive description logics such as $\mathcal{ALCNH}_{R^+}$ (Haarslev and Möller 2000) might lead to undecidable inference problems. In (Haarslev, Möller and Wessel 2001) it has been shown that $\mathcal{ALCNH}_{R^+}$ extended by a limited form of concrete domains leads to decidable inference problems. This is achieved by disallowing so-called feature chains in $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$. It is easy to see that the same pragmatic approach can also be applied to very expressive DLs. By analogy to $\mathcal{ALCNH}_{R^+}(\mathcal{D})^-$ the description logic $\mathcal{ALCQHI}_{R^+}(\mathcal{D})^-$ extends the logic $\mathcal{ALCQHI}_{R^+}$ or $\mathcal{SHIQ}$ (Horrocks, Sattler and Tobies 2000) with concrete domains.

An important property of the language $\mathcal{SHIQ}$ is that the subsumption hierarchy of the *TBox* part $\mathcal{T}$ of a *knowledge base* ($\mathcal{T}$, $\mathcal{A}$) is stable w.r.t. additions to the *ABox* part $\mathcal{A}$ (i.e. subsumption relations between concepts cannot be introduced by adding assertions to the ABox). In case of multiple knowledge bases ($\mathcal{T}$, $\mathcal{A}_1$), ..., ($\mathcal{T}$, $\mathcal{A}_n$), for query answering on any of the ABoxes $\mathcal{A}_i$ one can reuse computations done so far for the TBox $\mathcal{T}$ (e.g. indexing computations). This is due to the stability of the subsumption relationships between concepts, since they depend only on axioms in the TBox $\mathcal{T}$. This important property is lost when introducing *nominals*, which are described in the next subsection.

Very Expressive Description Logics

A *nominal* (the letter $\mathcal{O}$ in a language name indicates the presence of nominals) is a singleton concept, syntactically represented as $\{o\}$ and semantically interpreted as $\{\mathrm{o}\}^{\mathcal{I}} = \{o^{\mathcal{I}}\}$. Thus, *nominals* stand for concepts with exactly one individual in contrast to concepts which stand for a set of individuals. This allows the use of individuals in concept definitions, for instance, as names for specific persons, countries, etc., leading to the situation in which there is no longer a difference between TBoxes and ABoxes. OWL DL is a language that supports nominals.

$\mathcal{SROIQ}$ (Horrocks, Kutz and Sattler 2006) is one of the most expressive DL languages whose decidability has been proved. On top of $\mathcal{SHIQ}$ plus nominals, $\mathcal{SROIQ}$ allows for more expressivity concerning roles, where besides a TBox and an ABox, an *RBox* is introduced to include role statements, allowing for:

1. *Complex role inclusion axioms* of the form $R \circ S \sqsubseteq R$ and $S \circ R \sqsubseteq R$ where R is a role and S is a simple role.
2. *Disjoint roles*
3. *Reflexive, irreflexive and antisymmetric roles*
4. *Negated role assertions*
5. *Universal role*
6. *Local expressivity* to allow concepts of the form $\exists R.$Self

$\mathcal{SROIQ}$ represents the logical basis of OWL 1.1 plus datatypes and datatypes restrictions $\mathcal{SROIQ}(D^+)$. In (Horrocks et al. 2006) a tableaux algorithm is presented, proving that $\mathcal{SROIQ}$ is decidable if some restrictions concerning the so-called cyclicity of role axioms are obeyed.

As observed in previous sections, decidability is a characteristic that should be preserved by ontology languages and which has caused expressivity restrictions. This is one of the reasons why rules are gaining interest as an option to overcome expressivity limitations in DLs.

A relevant proposal to extend DL languages (more specifically, the syntactic variant OWL-DL) with rules, is the rule language called SWRL (Semantic Web Rule Language). SWRL uses OWL DL or OWL Lite as the underlying DL language to specify a KB. The syntax of SWRL is also based on XML. For brevity, however, we prefer a mathematical notation and define a rule as an axiom of the form

$$P_1(X_1, \ldots, X_{n_1}), \ldots, P_k(X_{n_1}, \ldots, X_{n_k}) \leftarrow \\ Q_1(Y_{1,1}, \ldots, Y_{1,m_1}), \ldots, Q_j(Y_{j,1}, \ldots, Y_{j,m_j})$$

such that P_{i_k} and Q_{i_j} are names and X_i mentioned in the head (lefthand side of the $\leftarrow$ constructor) as well as $Y_{i,j}$ (in the body on the righthand side) stand for variable names (or variables for short). Variables in the head must also be mentioned in the body. Predicate terms in a rule body are called (rule) atoms.

The semantics of SWRL rules is defined as follows. An interpretation satisfies a rule of the above form if it satisfies the first-order predicate

$$\forall X_1, \ldots, X_{n_1}, Y_{1,1}, \ldots, Y_{1,m_1}, \ldots, Y_{j,1}, \ldots, Y_{j,m_j} : \\ Q_1(Y_{1,1}, \ldots, Y_{1,m_1}) \land \ldots \land Q_j(Y_{j,1}, \ldots, Y_{,m_j}) \rightarrow \\ P_1(X_1, \ldots, X_{n_1}) \land \ldots \land P_k(X_{n_1}, \ldots, X_{n_k})$$

The extension of OWL DL with SWRL rules is known to be undecidable if predicate names are mentioned in the ontology (TBox) (Motik, Sattler and Studer 2005). Various decidable fragments of OWL DL with SWRL rules exist.

In order to add rules and still preserve decidability, a variant of SWRL can be used, the so called *DL-safe* rules (Motik et al. 2005). DL-safe rules are rules of the above form and are formally defined as follows.

Suppose a set of concept names N_c, a set of abstract and concrete role names $N_{R_a} \cup N_{R_c}$. A DL atom is of the form C(x) or R(x,y), where $C \in N_C$ and $R \in N_{R_a} \cup N_{R_c}$. Rule atoms may be DL atoms or atoms as defined above. A rule *r* is called safe if each of its variables also occur in a non-DL atom in the rule body. All rules must be safe. Additionally, in the ABox, assertions of the following form are allowed: $P(ind_1, \ldots, ind_n)$ where P is a name for a predicate used in a non-DL atom. The assertions are called *facts* (rules with empty bodies). Thus, in practice, the safety restriction introduced for DL-safe rules means that rules are applied to ABox individuals only. Note, however, that DL-safe rules are not trigger rules, they have a first-order semantics (and hence, e.g. the law of contraposition holds etc.).

In order to support the recognition of events in image sequences (see Section 3.4), rules with time variables will be used as part of a specific query language (see Section 2.2). We assume that assertions involving time variables such as, e.g. "ind_1 approaching ind_2 from t_1 to t_2" are generated by low-level image sequence analysis processes. The results are added to an ABox as so-called temporal propositions.

A *temporal proposition* is a syntactic structure of the following form:

$$P_{[t_1,t_2]}(ind_1, \ldots, ind_n)$$

where t_i denotes an element of a linear temporal structure $\Theta \subseteq \mathbb{N}$, ind_i with $i \in \{1, \ldots, n\}$ denotes an individual, and $P \in Preds$.

The semantics for rules with time intervals is different from DL-safe rules, and formally defined as follows. Let $\Theta \subseteq \mathbb{N}$ be a linear temporal structure. A temporal interpretation $\mathcal{I}_T$ is a tuple $(\Delta, \cdot^{\mathcal{I}}, \Theta, \Im)$ such that, in addition to the standard components of an interpretation, $\Im$ is an injective mapping from the temporal structure Θ to a set of standard Tarskian interpretation functions as used in previous sections.

A temporal interpretation $\mathcal{I}_T = (\Delta, \cdot^{\mathcal{I}}, \Theta, \Im,)$ satisfies a GCI or an ABox assertion if the standard part $(\Delta, \cdot^{\mathcal{I}})$ satisfies the GCI or the ABox assertion. The remaining components are used for defining satisfiability of temporal propositions. A temporal interpretation $\mathcal{I}_T$ satisfies a temporal proposition $P_{[t_1,t_2]}(ind_1, \ldots, ind_n)$ if the predicate is true for all time points in the non-empty interval $[t_1, t_2]$. Hence, we assume that temporal propositions are durative, i.e., the proposition holds for all non-empty subintervals (cf. (Neumann and Novak 1983) for a more detailed analysis). More formally:

$$\mathcal{I}_T \models P_{[t_1,t_2]}(ind_1, \ldots ind_n)$$

if for all $\theta \in \Theta$, $|\Theta| > 1$, it holds that if $t_1 \leq \theta \leq t_2$, then $(ind_1^{\Im(\theta)}, \ldots, ind_n^{\Im(\theta)}) \in P^{\Im(\theta)}$. As usual, a temporal interpretation that satisfies a temporal proposition is called a *temporal model* for this term. A temporal interpretation which satisfies a GCI or an ABox assertion is called a (temporal) model for the GCI or ABox assertion, respectively. An Abox with a set of temporal propositions such as, e.g. $\{move\_forward_{[10,20]}(ind_1), move\_backward_{[10,20]}(ind_1)\}$ should be inconsistent, but this requires (TBox) knowledge about the disjointness of predicates $move\_forward$ and $move\_backward$ for all time points. We ignore these issues here.

Temporal propositions are relevant for queries with time variables, which are described in Section 2.2.

2.2 Introduction to basic reasoning problems

Standard Inference Services

In the following we define standard inference services for description logics.

A *concept* C is called *consistent* (w.r.t. a TBox $\mathcal{T}$) if there exists a model of C (that is also a model of $\mathcal{T}$ and $\mathcal{R}$). An *ABox* $\mathcal{A}$ is *consistent* (w.r.t. a TBox $\mathcal{T}$) if $\mathcal{A}$ has model $\mathcal{I}$ (which is also a model of $\mathcal{T}$). A *knowledge base* ($\mathcal{T}$, $\mathcal{A}$) is called *consistent* if there exists a model for $\mathcal{A}$ which is also a model for $\mathcal{T}$. A concept, ABox, or knowledge base that is not consistent is called *inconsistent*.

A concept D *subsumes* a concept C (w.r.t. a TBox $\mathcal{T}$) if $\mathsf{C}^{\mathcal{I}} \subseteq \mathsf{D}^{\mathcal{I}}$ for all interpretations $\mathcal{I}$ (that are models of $\mathcal{T}$). If D subsumes C, then C is said to be *subsumed by* D.

For the definitions above, corresponding decision problems are defined as usual. In order to solve these problems, practical description logic systems implement algorithms as so-called inference services. Besides services for the basic decision problems introduced above, DL inference servers usually provide some additional inference services. A basic reasoning service is to compute the subsumption relationship between every pair of concept names mentioned in a TBox (i.e. elements from C). This inference is needed to build a hierarchy of concept names w.r.t. specificity. The problem of computing the most-specific concept names mentioned in $\mathcal{T}$ that subsume a certain concept is known as computing the *parents* of a concept. The *children* are the most-general concept names mentioned in $\mathcal{T}$ that are subsumed by a certain concept. We use the name *concept ancestors* (*concept descendants*) for the transitive closure of the parents (children) relation. The computation of the parents and children of every concept name is also called *classification* of the TBox. Another important inference service for practical knowledge representation is to check whether a certain concept name occcurring in a TBox is inconsistent. Usually, inconsistent concept names are the consequence of modelling errors. Checking the consistency of all concept names mentioned in a TBox without computing the parents and children is called a TBox *coherence check*.

If the description logic supports full negation, consistency and subsumption can be mutually reduced to each other since D subsumes C (w.r.t. a TBox $\mathcal{T}$) iff $\mathsf{C} \sqcap \neg\mathsf{D}$ is inconsistent (w.r.t. $\mathcal{T}$), and C is inconsistent (w.r.t. $\mathcal{T}$) iff C is subsumed by $\perp$ (w.r.t. $\mathcal{T}$). Consistency of concepts can be reduced to ABox consistency as follows: A concept C is consistent (w.r.t. a TBox $\mathcal{T}$) iff the ABox $\{\mathsf{a}:\mathsf{C}\}$ is consistent (w.r.t. $\mathcal{T}$).

An individual i is an *instance* of a concept C (w.r.t. a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$) iff $\mathsf{i}^{\mathcal{I}} \in \mathsf{C}^{\mathcal{I}}$ for all models $\mathcal{I}$ (of $\mathcal{T}$ and $\mathcal{A}$). For description logics that support full negation for concepts, the instance problem can be reduced to the problem of deciding if the ABox $\mathcal{A} \cup \{\mathsf{i}:\neg\mathsf{C}\}$ is inconsistent (w.r.t. $\mathcal{T}$). This test is also called *instance checking*. The most-specific concept names mentioned in a TBox $\mathcal{T}$ that an individual is an instance of are called the *direct types* of the individual w.r.t. a knowledge base $(\mathcal{T}, \mathcal{A})$. The direct type inference problem can be reduced to subsequent instance problems (see e.g. (Baader, Franconi, Hollunder, Nebel and Profitlich 1994) for details).

An ABox $\mathcal{A}'$ is *entailed* by a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ if all models of $\mathcal{T}$ and $\mathcal{A}$ are also models of $\mathcal{A}'$. For ABox entailment we write $\mathcal{T} \cup \mathcal{A} \models \mathcal{A}'$.

ABox entailment can be reduced to query answering. An ABox $\mathcal{A}'$ is entailed by a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ if for all assertions α in $\mathcal{A}'$ it holds that the boolean query $\{() \mid \alpha\}$ returns $true$. Query answering is discussed in the next subsection.

TBox inference services are provided by the systems CEL (Baader, Lutz and Suntisrivaraporn 2006), Fact++ (Tsarkov and Horrocks 2006), KAON2 (Hustadt, Motik and Sattler 2004), Pellet (Sirin and Parsia 2006), QuOnto (Calvanese, De Giacomo, Lembo, Lenzerini and Rosati 2005), and RacerPro (Haarslev and Möller 2001). At the time of this writing, only the latter four systems also support ABox inferences services.

Retrieval Inference Services

For practical applications, another set of inference services deals with finding individuals (or roles) that satisfy certain conditions.

The *retrieval* inference problem is to find all individuals mentioned in an ABox that are instances of a certain concept C. The set of *fillers* of a role R for an individual i w.r.t. a knowledge base $(\mathcal{T},\mathcal{A})$ is defined as $\{\mathsf{x} \mid (\mathcal{T},\mathcal{A}) \models (\mathsf{i},\mathsf{x}):\mathsf{R}\}$ where $(\mathcal{T},\mathcal{A}) \models ax$ means that all models of $\mathcal{T}$ and $\mathcal{A}$ also satisfy ax. The set of *roles* between two individuals i and j w.r.t. a knowledge base $(\mathcal{T},\mathcal{A})$ is defined as $\{\mathsf{R} \mid (\mathcal{T},\mathcal{A}) \models (\mathsf{i},\mathsf{j}):\mathsf{R}\}$.

In practical systems such as RacerPro, there are some auxiliary queries supported: retrieval of the concept names or individuals mentioned in a knowledge base, retrieval of the set of roles, retrieval of the role parents and children (defined analogously to the concept parents and children, see above), retrieval of the set of individuals in the domain and in the range of a role, etc. As a distinguishing feature to other systems, which is important for many applications, we would like to emphasize that RacerPro supports multiple TBoxes and ABoxes. Assertions can be added to ABoxes after queries have been answered. In addition, RacerPro and Pellet also provide support for retraction of assertions in particular ABoxes. The system Pellet can reuse previous computations.

Grounded conjunctive queries

In addition to the basic retrieval inference service described above (*concept-based instance retrieval*), more expressive query languages are required in practical applications. Well-established is the class of conjunctive queries.

A *conjunctive query* consists of a *head* and a *body*. The head lists variables for which the user would like to compute bindings. The body consists of query atoms (see below) in which all variables from the head must be mentioned. If the body contains additional variables, they are seen as existentially quantified. A query answer is a set of tuples representing bindings for variables mentioned in the head. A query is written $\{(X_1,\ldots,X_n) \mid atom_1,\ldots,atom_m\}$.

Query atoms can be *concept* query atoms ($C(X)$), *role* query atoms ($R(X,Y)$), *same-as* query atoms ($X = Y$) as well as so-called *concrete domain* query atoms. The latter are introduced to provide support for querying the concrete domain part of a knowledge base and will not be covered in detail here.

In the literature (e.g. (Horrocks, Sattler, Tessaris and Tobies 2000; Glimm, Horrocks, Lutz and Sattler 2007; Wessel and Möller 2006)), two different semantics for these kinds of queries are discussed. In *standard* conjunctive queries, variables (in the head and in query atoms in the body) are bound to (possibly anonymous) domain objects. A system supporting (unions of) grounded conjunctive queries is QuOnto.

In so-called *grounded* conjunctive queries, variables are bound to named domain objects (object constants). However, in grounded conjunctive queries the standard semantics can be obtained (only) for so-called tree-shaped queries by using existential restrictions in query atoms. Due to space restrictions, we cannot discuss the details here. In the following, we consider only (unions of) grounded conjunctive queries, which are supported by KAON2, RacerPro, and Pellet.

Complex queries are built from query atoms using boolean constructs for conjunction (indicated with comma), union ($\vee$) and negation ($\backslash$). Note that the latter refers to atom negation not concept negation and, for instance, negation as failure semantics is assumed in (Wessel and Möller 2005). In addition, a *projection* operator π is supported in order to reduce the dimensionality of an intermediate tuple set. This operator is particularly important in combination with negation (complement). These operators are only supported by RacerPro (for details see (Wessel and Möller 2005)).

In practical applications it is advantageous to name subqueries for later reuse, and practical systems, such as for instance RacerPro, support this for grounded conjunctive queries with non-recursive rules of the following form.

$$
\begin{aligned}
P(X_1, \ldots, X_{n_1}) \leftarrow\ & A_1(Y_1),\\
& \ldots\\
& A_l(Y_l),\\
& R_1(Z_1, Z2),\\
& \ldots\\
& R_h(Z_{2h-1}, Z_{2h}).
\end{aligned}
$$

The predicate term to the left of $\leftarrow$ is called the head and the rest is called the body, which, informally speaking, is seen as a conjunction of predicate terms. All variables in the head must be mentioned in the body, and rules must be non-recursive (with the obvious definition of non-recursivity). Since rules must be non-recursive there is no need to specify the semantics of rules because subsequent replacements (with well-known variable substitutions and variables renaming) of query atoms with their rule-defined body is possible (unfolding). For instance, unfolding an atom $P(X_1, \ldots, X_{n_1})$ results in a term $\pi(X_1, \ldots, X_{n_1}) : A_1(Y_1), \ldots A_l(Y_l), R_1(Z_1, Z2), \ldots R_h(Z_{2h-1}, Z_{2h})$. If there are multiple rules (definitions) for the same predicate P, corresponding disjunctions are generated. We do not discuss these details here, however.

It should be noted that answering queries in DL systems goes beyond query answering in relational databases. In databases, query answering amounts to model checking (a database instance is seen as a model of the conceptual schema). Query answering w.r.t. TBoxes and ABoxes must take all models into account, and thus requires deduction. The aim is to define expressive but decidable query languages. Well known classes of queries such as *conjunctive queries* and *unions of conjunctive queries* are topics of current investigations in this context.

A tuple $(ind_1, \ldots, ind_n)$ is in the result set of a grounded conjunctive query

$$
\begin{aligned}
\{(X1, \ldots, X_n) \mid\ & A_1(Y_1),\\
& \ldots\\
& A_l(Y_l),\\
& R_1(Z_1, Z2),\\
& \ldots\\
& R_h(Z_{2h-1}, ZW_{2h})\}
\end{aligned}
$$

if the variable substitution $[X_1 \leftarrow ind_1, \ldots, X_n \leftarrow ind_n]$ can be extended such that additional substitutions for all other variables in the body can be found such that the resulting query atoms after applying the substitution are satisfied in all models of the ontology (TBox and ABox). Hence, given a variable substitution, grounded conjunctive queries can be reduced to standard inference problems, which are discussed above. For unions and projections, the semantics is slightly more complicated, and we refer to (Wessel and Möller 2006). Although, for brevity, in this chapter we use a mathematical notation for conjunctive queries, there exist proposals for conjunctive queries in the XML-based DIG 2.0 format (Turhan, Bechhofer, Kaplunova, Liebig, Luther, Möller, Noppens, Patel-Schneider, Suntisrivaraporn and Weithöner 2006). In addition, another XML-based format called OWL-QL has been proposed, and practical query answering systems are available (e.g. (Kaplunova, Kaya and Möller 2006)).

Queries w.r.t. temporal propositions:

In order to support event recognition in an ontology-based media interpretation system, we introduced temporal propositions. For queries over ABoxes that also contain temporal propositions, rules with time intervals can be defined. Suppose three disjoint sets of names $Preds$, $TimeVars$ and $Vars$ neither of which is a subset of the names mentioned in the axioms of the ontology. Then, a *rule with time intervals* has the following structure:

$$\begin{aligned}
P_{[T_0,T_1]}(X_1,\ldots,X_{n_1}) \leftarrow\ & Q_{1[T_2,T_3]}(Y_{1,1},\ldots,Y_{1,m_1}),\\
& \ldots\\
& Q_{k[T_{2k},T_{2k+1}]}(Y_{k,1},\ldots,Y_{k,m_k}),\\
& A_1(Z_1),\\
& \ldots\\
& A_l(Z_l),\\
& R_1(W_1,W2),\\
& \ldots\\
& R_h(W_{2h-1},W_{2h}).
\end{aligned}$$

where the $T_i \in TimeVars$ are temporal variables and X_i, $Y_{j,k}, Z_l, W_h \in Vars$ are (not necessarily disjoint) variables that are bound to individuals mentioned in the ABox, $P, Q_i \in Preds$, and A_j, R_k are concept names and role names, respectively. In a similar way as for conjunctive queries introduced above, all variables in the head must be mentioned in the body, and rules must be non-recursive. Thus, queries w.r.t. time variables are unfolded, similar to rules for defined queries.

A *conjunctive query with time variables* is an expression of the following form:

$$\begin{aligned}
\{(X_1,\ldots,X_n)_{[T_1,T_2]} \mid\ & Q_{1[T_2,T_3]}(Y_{1,1},\ldots,Y_{1,m_1}),\\
& \ldots\\
& Q_{k[T_{2k},T_{2k+1}]}(Y_{k,1},\ldots,Y_{k,m_k}),\\
& A_1(Z_1),\\
& \ldots\\
& A_l(Z_l),\\
& R_1(W_1,W2),\\
& \ldots\\
& R_h(W_{2h-1},W_{2h})\}
\end{aligned}$$

Note that the variables $X_i, Y_{i,j}, Z_i$, and W_i are not necessarily disjoint. A tuple $(ind_1, \ldots, ind_n)_{[t_1,t_2]}$ is a potential solution of a grounded unfolded temporal conjunctive query (temporal query for short) if the variable substitution $[X_1 \leftarrow ind_1, \ldots, X_n \leftarrow ind_n, T_1 \leftarrow t_1, T_2 \leftarrow t_2]$ can be extended with additional assignments for all other variables in the body such that the resulting query atoms after applying the substitution are satisfied in all temporal models of the ontology (TBox and ABox). The result set for a temporal query comprises all tuples $(ind_1, \ldots, ind_n)_{[(t_{1_{min}}, t_{1_{max}}),(t_{2_{min}}, t_{2_{max}})]}$ such that there exists no other potential solution $(ind_1, \ldots, ind_n)_{[t_1,t_2]}$ with $t_1 < t_{1_{min}}$ or $t_1 > t_{1_{max}}$ or $t_2 < t_{2_{min}}$ or $t_2 > t_{2_{max}}$.

Algorithms for answering queries involving rules with time variables have been published in (Neumann and Novak 1983) and (Neumann 1985). The algorithms are implemented as inferences only in the RacerPro description logic system. In addition to the original Prolog-style approach in (Neumann 1985), conjunctive query atoms for ABoxes are provided for queries with time variables.

Nonstandard Inference Services

Many inference services different from those mentioned above have been introduced in the literature (non-standard inference services). We discuss only one non-standard inference service, namely abduction, which is relevant for the media interpretation processes described below (Elsenbroich, Kutz and Sattler 2006).

The *abduction* inference service aims to construct a set of (minimal) explanations Δ for a given set of assertions Γ such that Δ is consistent w.r.t. to the ontology $(\mathcal{T}, \mathcal{A})$ and satisfies:

1. $\mathcal{T} \cup \mathcal{A} \cup \Delta \models \Gamma$ and
2. If Δ' is an ABox satisfying $\mathcal{T} \cup \mathcal{A} \cup \Delta' \models \Gamma$, then $\Delta' \models \Delta$ (Δ is least specific)

This inference service is used in Section 3 as the basis for formalizing the derivation of annotations (metadata) for media objects. The annotations describe high-level interpretations of media objects. Furthermore, they can be used to retrieve the media objects from which they are derived. Often, the names in Δ and Γ are predefined (and are called abducibles and observables, respectively).

2.3 Retrieval of media objects

An application scenario for automatically derived interpretations of media objects is information retrieval, for instance, in the Semantic Web. Interpretations are seen as annotations of media objects and can be practically represented in RDF or OWL format. In our view, annotations describe “real-world” objects and events. It is not the goal to merely “classify” images and attach keywords but to construct a high-level interpretation of the content of a media object. The former approach has a limited applicability if examples such as Figure 3.1 are considered and queries for, e.g. media objects with a mailman have to be answered. The goal of this section is to motivate the use of Aboxes for describing media content in constrast to using just keywords (or concept names) for classifying media objects. Details about how description logics can be used for media retrieval based on description logics have been published in (Möller, Haarslev and Neumann 1998), see also subsequent work in (Di Sciascio, Donini and Mongiello 1999; Di Sciascio, Donini and Mongiello 2000; Schober, Hermes and Herzog 2005). In a more general setting, (Sebastiani 1994) deals with description logic and information retrieval.

A set of media objects with annotations attached to each media object can be made available via a web server with standard application server technology. We assume that the web server provides a query interface (for instance, using the XML-based DIG 2.0 or OWL-QL query language, see Section 2.2). For readability reasons, however, here we use ABoxes for content descriptions, and employ a mathematical notation for queries. Details about XML-based multimedia content descriptions and MPEG-7 have been described in Chapters 2 and 3 of this book.

$$
\begin{array}{rcl}
mailman_1 & : & Mailman \\
bicycle_1 & : & Bicycle \\
mail\_deliv_1 & : & MailDeliv \\
(mail\_deliv_1, mailman_1) & : & hasPart \\
(mail\_deliv_1, bicycle_1) & : & hasPart \\
(mail\_deliv_1, url\_1) & : & hasURL \\
(mailman_1, url\_2) & : & hasURL \\
(bicycle_1, url\_3) & : & hasURL \\
(url\_1) & : & =_{\texttt{"http://www.img.de/image-1.jpg"}} \\
(url\_2) & : & =_{\texttt{"http://www.img.de/image-1.jpg\#(200,400)/(300/500)"}} \\
(url\_3) & : & =_{\texttt{"http://www.img.de/image-1.jpg\#(100,400)/(150/500)"}} \\
garbageman_1 & : & Garbageman \\
garbageman_2 & : & Garbageman \\
garbagetruck_1 & : & Garbage\_Truck \\
gc_1 & : & Garbage\_Collection \\
(gc_1, garbageman_1) & : & hasPart \\
(gc_1, garbageman_2) & : & hasPart \\
(gc_1, garbagetruck_1) & : & hasPart \\
(gc_1, url\_4) & : & hasURL \\
 & \dots &
\end{array}
$$

Fig. 3.6. An ABox representing the annotation of the image in Figure 3.1. The predicate $=_{string}$ stands for a one-place predicate $p(x)$ which is true for $x = string$.

$$
\begin{aligned}
ImageQuery_1 &:= \{(X, Y) \mid MailDeliv(X), Bicycle(Y), hasPart(X, Y)\} \\
URLQuery_1 &:= \{(X, value(X)) \mid hasURL(mail\_deliv_1, X)\} \\
URLQuery_2 &:= \{(X, value(X)) \mid hasURL(bicycle_1, X)\}
\end{aligned}
$$

Fig. 3.7. Query for “a mail delivery with a bicycle” and subsequent queries for retrieving the URLs w.r.t. the result for $ImageQuery_1$.

Using the example from Figure 3.1 we sketch how media interpretations are used to implement a media retrieval system. Figure 3.6 illustrates the main ideas about annotations for media objects using ABoxes (we omit the TBox for brevity). It would have been possible to more appropriately describe the role which the parts play in the events (in the sense of case frames). We omit the discussion of these issues here for brevity, however, and use a "generic" role $hasPart$. It is also possible to use another "aggregate" $street\_scene_1$ for combining the garbage collection and mail delivery events.

A query which might be posed in an information system is shown in Figure 3.7. As a result, the inference system returns the tuple $(mail\_deliv_1, bicycle_1)$, and in order to show the image (and highlight the area with the bicycle), the associated URL names can be retrieved (see also Figure 3.7). The form $value(x)$ returns a unique binding for a variable (in this case a string) if it exists, and $\emptyset$ otherwise. In case of $URLQuery_1$ the answer is $(url_1,$ `"http://www.img.de/image-1.jpg"`$)$. The result of $URLQuery_2$ is defined analogously. The URLs can be used to actually retrieve the image data. Subsequent queries w.r.t. the annotation individuals $mail\_deliv_1$ and $bicycle_1$ are certainly possible. We do not discuss details here, however. In summary, it should be clear now, how annotations with metadata are used in an ontology-based information retrieval system.

With axioms such as

$$Mailman \sqsubseteq Postal\_Employee$$

$$Mailman \equiv Postman$$

a query for a $Postal\_Employee$ or a $Postman$ will also return the media object shown in Figure 3.1. In general, all benefits of description logic reasoning carry over to query answering in an information retrieval system of the kind sketched above.

It is easy to see that annotations such as the ones shown in Figure 3.6 can be set up such that the URLs are tied to the ABox individuals comprising the high-level descriptions. In particular, one can easily imagine a situation in which there exist multiple interpretations of an image, which results in multiple annotations being associated with an image. In addition, it is obvious that a repository of media objects together with their annotations (metadata) gives rise to one or more ABoxes that are managed by the ontology-based information system. Not so obvious is how metadata can be automatically derived since manual annotation is too costly in almost all practical scenarios. The derivation of metadata representing high-level interpretation of media content is discussed in the next section. Querying as discussed in this section refers to metadata (Aboxes) but not directly to media content. Processes for the automatic derivation of metadata, however, do refer to media content, and as we will see later, a set of given queries can indeed influence media interpretation.

3 Automatic construction of metadata for media objects

In this section we will discuss how media objects can be automatically interpreted. We start with images, continue with text, and finally discuss image sequence interpretation.

A first attempt to understand fusion of information gained w.r.t. different modalities is presented afterwards.

3.1 Image interpretation

An ontology in a description logic framework is seen as a tuple consisting of a TBox and an ABox. In order to construct a high-level interpretation, the ABox part of the ontology is extended with some new assertions describing individuals and their relations. These descriptions are derived by media interpretation processes using the ontology (we assume the ontology axioms are denoted in a set Σ).

Interpretation processes are set up for different modalities, still images, videos, audio data, and texts. In this section we discuss the interpretation process using an example interpretation for still images. The output is a symbolic description represented as an ABox. This ABox is the result of an abduction process (see (Hobbs, Stickel, Appelt and Martin 1993; Shanahan 2005) for a general introduction). In this process a solution for the following equation is computed: $\Sigma \cup \Delta \models \Gamma$. The solution Δ must satisfy certain side conditions (see Section 2.2).

Fig. 3.8. Still image displaying a pole-vault event.

In Figure 3.8 an example from the athletics domain is presented. Assuming it is possible to detect a horizontal bar bar_1, a human $human_1$, and a pole $pole_1$ by image analysis processes, the output of the analysis phase is represented as an ABox Γ. Assertions for the individuals and (some of) their relations detected by analysing Figure 3.8 are shown in Figure 3.9. We are aware of the fact that crisp object recognition might be hard to achieve. Therefore, in Section 4 we develop an approach that deals with uncertainty in this respect.

In order to continue the interpretation example, we assume that the ontology contains the axioms shown in Figure 3.10 (the ABox of the ontology is assumed to be

$$
\begin{aligned}
pole_1 &: Pole \\
human_1 &: Human \\
bar_1 &: Horizontal\_Bar \\
(bar_1, human_1) &: near \\
(human_1, pole_1) &: touches
\end{aligned}
$$

Fig. 3.9. An ABox Γ representing the result of the image analysis phase.

$$
\begin{aligned}
Man &\sqsubseteq Human \\
Woman &\sqsubseteq Human \\
Man &\sqsubseteq \neg Woman \\
Athlete &\equiv Human \sqcap \exists hasProfession.Sport \\
Jumper &\sqsubseteq Athlete \\
Foam\_Mat &\sqsubseteq SportEquipment \\
Pole &\sqsubseteq SportEquipment \\
Javelin &\sqsubseteq SportEquipment \\
Horizontal\_bar &\sqsubseteq SportEquipment \\
Jumping\_Event &\sqsubseteq Event \sqcap \\
&\quad \exists hasPart.Jumper \sqcap \\
&\quad \exists_{\leq 1} hasPart.Jumper \\
Pole\_Vault &\sqsubseteq Jumping\_Event \sqcap \\
&\quad \exists hasPart.Pole \sqcap \\
&\quad \exists hasPart.Horizontal\_Bar \sqcap \\
&\quad \exists hasPart.Foam\_Mat \\
High\_Jump &\sqsubseteq Jumping\_Event \sqcap \\
&\quad \exists hasPart.Horizontal\_Bar \sqcap \\
&\quad \exists hasPart.Foam\_Mat \\
PV\_InStartPhase &\sqsubseteq \top \\
PV\_InEndStartPhase &\sqsubseteq \top \\
HJ\_InJumpPhase &\sqsubseteq \top \\
&\ldots
\end{aligned}
$$

Fig. 3.10. An tiny example TBox Σ for the athletics domain.

empty). If we compare with the aggregate design patterns shown in Figure 3.4, axioms for both, $Pole\_Vault$ as well as $High\_Jump$, contain parent concepts and restrictions for parts. However, in Figure 3.10 there are no constraints between part objects. Therefore, the conditions mentioned on the right-hand side are only necessary and not sufficient conditions as in Figure 3.4. For expressing constraints between parts in an aggregate (at least three objects are involved), description logics are not expressive enough (only the two-variable fragment of first order logic is captured). Thus, some additional mechanism is required without jeopardizing decidability. In order to capture constraints among aggregate parts, we assume that the ontology is extended with DL-safe rules (rules that are applied to ABox individuals only, see Section 2.1). In Figure 3.11 a set of rules for the athletics example is specified. Note that the spatial constraints $touches$ and $near$ for the parts of a $Pole\_Vault$ event (or a $High\_Jump$ event) are not imposed by the TBox in Figure 3.10. Thus, rules are used to represent additional knowledge. Since spatial relations depend on the specific "subphases" of the events, corresponding clauses are included on the right-hand sides of the rules. For instance, a jumper as part of a $High\_Jump$ is near the bar if the image shows a $High\_Jump$ in the jump phase. Later, in the context of fusion discussed in Section 3.5, we will see how information about the phase (e.g. $HJ\_InJumpPhase$) as captured in an image is related to the spatio-temporal knowledge of the image sequence modality.

In the following we assume that rules such as those shown in Figure 3.11 are part of the TBox Σ.

$$
\begin{aligned}
touches(Y,Z) \leftarrow\ & Pole\_Vault(X), \\
& PV\_InStartPhase(X), \\
& hasPart(X,Y), Jumper(Y), \\
& hasPart(X,Z), Pole(Z). \\
near(Y,Z) \leftarrow\ & Pole\_Vault(X), \\
& PV\_InEndStartPhase(X), \\
& hasPart(X,Y), Horizontal\_Bar(Y), \\
& hasPart(X,Z), Jumper(Z). \\
near(Y,Z) \leftarrow\ & High\_Jump(X), \\
& HJ\_InJumpPhase(X), \\
& hasPart(X,Y), Horizontal\_Bar(Y), \\
& hasPart(X,Z), Jumper(Z). \\
\ldots &
\end{aligned}
$$

Fig. 3.11. Additional restrictions for $Pole\_Vault$ and $High\_Jump$ in the form of rules.

In order to provide a high-level interpretation, i.e. to provide a description of the image content in the form of high-level aggregates, we assume that spatial relations between certain objects detected by low-level analysis processes are not arbitrary. In order to construct an interpretation, an explanation is computed why it is the case that a jumper touches a pole and is near a horizontal bar. Such explanations are considered the results of image interpretation processes. As mentioned above, the idea is to use the abduction inference service for deriving these kinds of (minimal) explanations (in the sense of interpretations). Minimal explanations can be extended appropriately in order to match expectations and task context.

We start with the computation of a minimal explanation in our athletics scenario. For this purpose, we slightly modify the abduction equation by taking into consideration that initially the ABox does not need to be empty. Thus, we divide Γ (see Figure 3.9) into a part Γ_2 that the agent would like to have explained, and a part Γ_1 that the interpretation agent takes for granted. In our case Γ_2 is $\{(bar_1, human_1) : near,\quad (human_1, pole_1) : touches\}$ and Γ_1 is $\{pole_1 : Pole,\quad human_1 : Human,\quad bar_1 : Horizontal\_Bar\}$.

Coming back to the abduction problem specified above, we need solution(s) for the equation $\Sigma \cup \Delta \cup \Gamma_1 \models \Gamma_2$. In other words, given the background ontology Σ from Figures 3.10 and 3.11, a query as derived from Γ_2 should return $true$ (see Figure 3.12).

Obviously, this is not the case if Δ is empty. In order to see how an appropriate Δ could be derived, let us have a look at the rules in Figure 3.11. In particular, let

$$Q_1 := \{() \mid near(bar_1, human_1),\\ touches(human_1, pole_1)\}$$

Fig. 3.12. Query representing Γ_2.

us focus on the rules for $Pole\_Vault$ first. If we apply the rules to the query in a backward chaining way (i.e. from left to right) and unify corresponding terms we get variable bindings for Y and Z. The "unbound" variable X of the corresponding rules is instantiated with fresh individuals (e.g. pv_1 and pv_2). Since the parts and their relations can be explained with one aggregate, it is reasonable to assume that only one event provides a complete explanation, i.e. only one individual pv_1 is used (Occam's Razor). Then, a possible solution Δ for the abduction equation can be derived. Δ is shown in Figure 3.13.

$$\begin{aligned} pv_1 &: Pole\_Vault \\ pv_1 &: PV\_InStartPhase \\ pv_1 &: PV\_InEndStartPhase \\ human_1 &: Jumper \\ (pv_1, human_1) &: hasPart \\ (pv_1, bar_1) &: hasPart \\ (pv_1, pole_1) &: hasPart \end{aligned}$$

Fig. 3.13. One possible solution of the abduction equation.

Note that due to the involvement of $human_1$ in the pole-vault event, $human_1$ is now seen as an instance of $Jumper$, and, due to the TBox, also as an $Athlete$. Thus, information from high-level events also influences information that is available about the related parts. With queries for $Jumper$s the corresponding media objects would not have been found otherwise. Thus, recognizing high-level events is of utmost importance in information retrieval systems (and pure content-based retrieval does not help).

Considering the GCIs involving $Pole\_Vault$ in the TBox shown in Figure 3.10 it becomes apparent that for a pole vault there also exists a foam mat which is not found by the image analysis module: Maybe it is not visible or the analysis just could not detect it. In the latter situation, one could somehow adapt the image analysis processes and start a feedback loop. This feedback from the image interpretation module (high level) to the image analysis module (low-level) is subject to ongoing research and will be covered in more detail in Section 3.6. The assertions concerning the relation $hasPart$ and the phases derived by the rule are included in the interpretation result. Thus, the output of the interpretation phase in our example is the ABox shown in Figure 3.14.

The example discussed here covers the interpretation of still images. It is necessary, however, to keep in mind that each media object might consist of multiple modalities, each of which will be the basis of modality-specific interpretation results (ABoxes). In order to provide for an integrated representation of the interpretation of media objects

$$
\begin{aligned}
pole_1 &: Pole \\
human_1 &: Human \\
bar_1 &: Horizontal\_Bar \\
(bar_1, human_1) &: near \\
(human_1, pole_1) &: touches \\
pv_1 &: Pole\_Vault \\
pv_1 &: PV\_InStartPhase \\
pv_1 &: PV\_InEndStartPhase \\
human_1 &: Jumper \\
(pv_1, human_1) &: hasPart \\
(pv_1, bar_1) &: hasPart \\
(pv_1, pole_1) &: hasPart
\end{aligned}
$$

Fig. 3.14. An ABox representing the result of the image interpretation phase.

as a whole, these modality-specific interpretation results must be appropriately integrated. A cornerstone of this integration process will be to determine which modality-specific names refer to the same domain object. This will be discussed in later sections. In the following, both modality-specific and media-specific ABoxes will be called interpretation ABoxes. In a specific context, ambiguities should not arise.

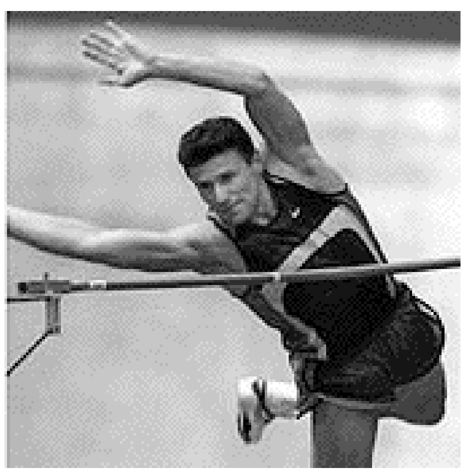

Fig. 3.15. Image displaying a snapshot of a high jump or pole vault (where the pole is outside the image).

So far we have discussed an example where there is one unique explanation (and, hence, one unique interpretation). However, this need not necessarily be the case. In Figure 3.15 an example is presented that might lead to two different interpretations. For the example we assume that the ABox in Figure 3.16 is produced by the image analysis component.

$$
\begin{aligned}
bar_2 &: Horizontal\_Bar \\
human_2 &: Human \\
(bar_2, human_2) &: near
\end{aligned}
$$

Fig. 3.16. An ABox Γ representing the result of the analysis of the image in Figure 3.15.

For the interpretation process we assume the same ontology as above. It is easy to see that we can get two explanations by the abduction process (see Figures 3.17 and 3.18). Note that new names which might refer to the same domain object are used in each explanation.

Continuing the example, it might be the case that for some images the ontology does not contain relevant axioms or rules. In this case, the interpretation result, i.e. the result of solving the abduction problem $\Sigma \cup \Delta \cup \Gamma_1 \models \Gamma_2$ will be degenerate because, due to missing axioms or rules in Σ, Δ must necessarily be equal to Γ_2 in order to solve the equation. As an example of such a situation we can discuss an interpretation of Figures 3.8 or 3.15 without the rules from Figure 3.11 and the GCIs for $Pole\_Vault$ and $High\_Jump$ in Figure 3.10. The degenerate interpretation result is shown (as Γ) in Figure 3.9. An annotation based on such a degenerate interpretation will certain not support queries such as $\{(x) \mid Pole\_Vault(x) \vee High\_Jump(x)\}$.

$$
\begin{aligned}
human_2 &: Human \\
bar_2 &: Horizontal\_Bar \\
(bar_2, human_2) &: near \\
hj_2 &: High\_jump \\
hj_2 &: HJ\_InJumpPhase \\
human_2 &: Jumper \\
(hj_2, human_2) &: hasPart \\
(hj_2, bar_2) &: hasPart
\end{aligned}
$$

Fig. 3.17. An ABox representing the first result of the image interpretation process.

$$
\begin{aligned}
human_2 &: Human \\
bar_2 &: Horizontal\_Bar \\
(bar_2, human_2) &: near \\
pv_2 &: Pole\_Vault \\
pv_2 &: PV\_InEndStartPhase \\
human_2 &: Jumper \\
(pv_2, human_2) &: hasPart \\
(pv_2, bar_2) &: hasPart
\end{aligned}
$$

Fig. 3.18. An ABox representing the second result of the image interpretation process.

In Figure 3.19 a pole vault is shown. Suppose the ABox shown in Figure 3.20 is generated by image analysis processes. Compared to Figure 3.16 there is only one

Fig. 3.19. Image displaying a snapshot of a pole vault (where the pole is partially outside the image).

$$\begin{aligned} bar_3 &: Horizontal\_Bar \\ jumper_3 &: Human \\ pole_3 &: Pole \\ (bar_3, jumper_3) &: near \end{aligned}$$

Fig. 3.20. An ABox Γ representing the result of the analysis of the image in Figure 3.19.

additional assertion, the assertion for the pole. If we apply the abduction process in a naive way, the result will also be two interpretation ABoxes as shown above (one for a pole vault and one for a high jump). In the high-jump event, the pole is just ignored (and erroneously considered as "noise"). As the abduction process is defined now, there is no reason to explain the pole since up to now only the spatial relations are put into Γ_2 and hence are "explained". The example demonstrates, that also assertions about single objects have to be put into Γ_2 in order to avoid spurious effects.

3.2 Towards an abduction procedure

The interpretation example presented so far exhibits several interesting characteristics which will now be discussed in greater generality.

First, it is important to note that, in general, interpretations do not logically follow from the data and the knowledge base. Visual or audio data are inherently ambiguous, and multiple interpretations may be possible. Hence deductive reasoning is not adequate. Rather, media data must be seen as a causal consequence of some real-world scenario which is to be described by an interpretation. For example, natural images are caused by projecting 3D scenes, and interpretations of the images should provide descriptions of the underlying 3D scenes. Furthermore, media data may be sparse, describing only parts of a scenario. For example, Figure 3.15 is just a snapshot of a complete high-jump occurrence. Obviously, sparse data may be ambiguous and interpreted in several ways, in this case as a high-jump or a pole-vault event.

In the example, the causal relationship between high-level concepts (such as $Pole\_Vault(X)$) and relations between low-level data (such as near(Y, Z)) is represented by rules because description logics are not expressive enough for these kinds of constraints. In some case, however, there might be axioms in the Tbox that provide necessary conditions (see, e.g. the axiom for Athlete in Figure 3.10). These axioms are more general than corresponding rules (they apply to all domain objects, not only to objects for which there is a name in the Abox). A rule rule such as

$$\begin{aligned} Athlete(X) \leftarrow\; & Human(X) \\ & hasProfession(X, Y) \\ & Sport(Y) \end{aligned}$$

might be implicitly derived from the axiom. This approximation process might take into account a set of externally-defined abducibles in order to limit the number of axioms to be considered for the abduction operation. Details of the approximation process are subject to further research. Using rules for sufficient conditions is one way to enable backward chaining from low-level data to high-level explanations. In general, such rules should also be available for all conjunctive constituents of a high-level concept in order to enable backward chaining along multiple paths. For instance, the following concept inclusion (see Figure 3.10)

$$\begin{aligned} High\_Jump \sqsubseteq\; & Jumping\_Event \sqcap \\ & \exists hasPart.Horizontal\_Bar \sqcap \\ & \exists hasPart.Foam\_Mat \end{aligned}$$

should give also rise to the rules

$$\begin{aligned} Jumping\_Event(X) \leftarrow\; & High\_Jump(X) \\ Horizontal\_Bar(X) \leftarrow\; & High\_Jump(Y), \\ & hasPart_{Bar}(Y, X). \\ Foam\_Mat(X) \leftarrow\; & High\_Jump(Y), \\ & hasPart_{Mat}(Y, X). \end{aligned}$$

where $hasPart_C$ represents a relation (or role) that associates a high-jump instance with a part instance of concept C (the role $hasPart_C$ is range-restricted to C). In this case there is no simple syntactic transformation for exploiting the GCI for $High\_Jump$ in the abduction process. A possible explanation for the data

$$bar_1 : Horizonal\_Bar$$

could be generated by the rule which states that a $Horizontal\_Bar$ can be part of a $High\_Jump$. Hence, a high-jump aggregate can be generated to which bar_1 is associated via the function $hasPart_{Bar}$. Similar arguments and rules derived from the Tbox will lead to a pole-vault aggregate as another (minimal) explanation for a bar in a picture (without the pole being shown). If, additionally, there is a pole as in Figure 3.19, and it is mentioned in the analysis ABox, e.g.

$$pole_1 : Pole$$

then, unfortunately, a high jump does provide an explantion for the bar (neglecting the pole). Thus if we require the process to explain the existence of objects, spurious interpretations such as “a high jump with some arbitrary pole” (see Figure 3.20) can be avoided. This has been discussed in the literature as the principle of consilience (Hobbs, Stickel, Martin and Edwards 1988; Hobbs et al. 1988; Hobbs, Stickel, Appelt and Martin 1990).

It is evident that interpretation by abduction can in general be achieved by exploiting the $hasPart$ structure of concepts and explaining data as part of a larger whole. It is therefore useful to view possible interpretation steps within the compositional hierarchy of aggregate concepts (Neumann and Weiss 2003). Figure 3.21 shows a compositional hierarchy for the domain of sports events, based on and extending the example TBox in Figure 3.10. Given the media data shown in Figure 3.8, the compositional hierarchy exposes all aggregate concepts which could explain the data.

In order to provide additional support for an explanation, it may also be useful to follow $hasPart$ arcs from aggregates to parts. For instance, to verify a pole vault, one may check whether the media data includes a pole.

Besides the compositional hierarchy, the taxonomical hierarchy can also be exploited for generating explanations. For example, $athlete_1 : Athlete$ may not have been recognized, and $moving\_object_1 : Person$ is included in the media data instead. In this case it proves useful to explore a possible specialization of $Person$ which would then lead to interesting aggregate concepts.

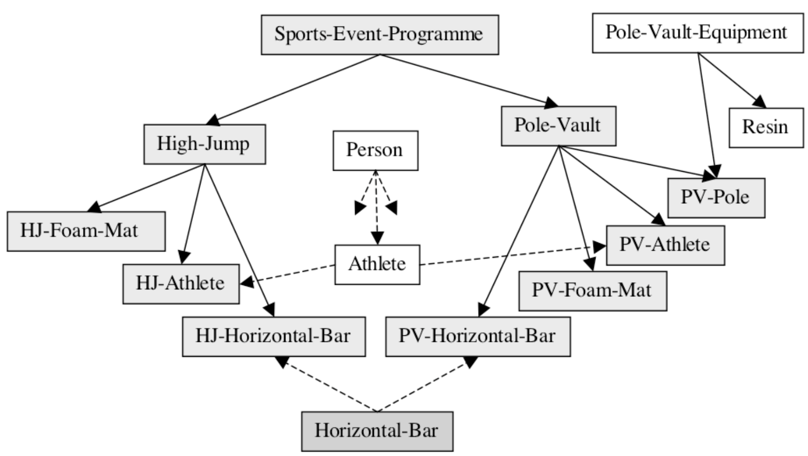

Fig. 3.21. Compositional hierarchy induced by hasPart roles in aggregate concepts. The dotted arrows indicate specialisation relations.

In summary, possible explanations can be generated by navigating from data to higher-level concepts by

- aggregate instantiation,
- aggregate expansion, and
- instance specialization.

These considerations can be used to guide the knowledge modelling process and might help to find appropriate rules for the abduction process described above.

As already shown by (Reiter and Mackworth 1990) and further elaborated in (Schröder 1999), image interpretation can also be formally described as constructing a partial model. "Model" is used here in the logical sense and means a mapping from the symbols of logical formulae into a real-world domain such that the formulae are true. A partial model can be constructed by a computer (which has no direct access to the real-world domain) by building the model on top of the primitive media data which are taken to map into the intended real-world objects. As opposed to interpretation by abduction, interpretation by model construction does not focus on the data but aims at constructing a symbolic description of some real-world scenario consistent with the data.

As pointed out in Section 1, the scope of an interpretation depends also on the task and other contextual information. However, the logical formalisation in terms of model construction gives no clue as to what to include and what not to include in an interpretation as long as the interpretation is consistent and entails the data. One way to restrict interpretations is by introducing a notion of dependency and by requiring that the interpretation should depend on the data. This can be formalised by considering the compositional hierarchy formed by aggregate concepts. We say that a concept depends on data D if it is a predecessor of some element of D or a successor of a predecessor. This notion of dependency is the same as in Bayesian Networks with causality arcs corresponding to the hasPart relations in the figure. In Figure 3.21, it is assumed that $Horizontal\_Bar$ is the data. All concepts dependent on $Horizontal\_Bar$ are depicted in light grey.

The dependency definition can be further refined by distinguishing between necessary and optional parts (expressed by cardinality restrictions for the $hasPart$ roles). If, for example, a sports event has all its parts as optional parts, it would make little sense to include a $High\_Jump$ as part of the explanation for a $Pole\_Vault$.

Restricting interpretations to assertions which in this sense depend on the data appears to be useful for many interpretation tasks. However, one can also conceive of tasks where non-dependent assertions may be interesting, for example providing further explanations for some of the hypothesized instances. In any case, a task-based control of the scope of an interpretation remains necessary.

3.3 Shallow text interpretation

Another modality which might provide additional information for media interpretation is “text”. There exists a large amount of publications about natural language understanding, text interpretation, information retrieval from text, etc. We cannot give a survey of all trends and current research results here. However, using an example, we show how the technique of abduction introduced in the previous section can be used to provide interpretation ABoxes. The goal of the example discussed in this subsection is to demonstrate the feasibility of the general approach for multimedia interpretation. Abduction for natural language interpretation is investigated in much more detail in

(Hobbs et al. 1988). Abduction is even used to formalise discourse understanding. No decidable representation formalism is used, however.

Our example assumes that standard techniques from information retrieval approaches are applied ("shallow text processing"). Consider the sentences "A new world record in this year's event was missed. The remaining famous athlete touched the crossbar and failed 2.40m." We assume that the ABox shown in Figure 3.22 is generated by low-level text analysis components. For the nouns, individuals are generated as instances of appropriate concepts (we suppose a mapping from word to concepts is taken from a gazetteer e.g. "crossbar" $\rightarrow$ $Horizontal\_Bar$). The role $precedes$ represents the fact that there exists a linear precedence between corresponding nouns across adjacent sentence boundaries. In our example, the athlete and the crossbar are mentioned in the sentence immediately after the sentence with the event.

$$\begin{aligned} object_1 &: Event \\ object_2 &: Horizontal\_Bar \\ object_3 &: Athlete \\ object_3 &: Famous \\ (object_1, object_2) &: precedes \\ (object_1, object_3) &: precedes \end{aligned}$$

Fig. 3.22. Interpretation ABox produced by shallow text analysis.

Note that in Figure 3.22 for the word "famous" there is no new object generated. The word "famous" is used as an adjective here, and this can be easily detected even by shallow text processing techniques. For generating an interpretation ABox we assume that the $precedes$ assertions are to be explained (Γ_2) whereas the first four assertions are taken for granted (Γ_1). The query

$$Q_2 := \{() \mid precedes(object_1, object_2), precedes(object_1, object_3)\}$$

represents Γ_2 in a similar way as discussed in Section 3.1. The goal is to compute a Δ that explains the "surface relation" $precedes$ with a "semantically deep relation" such has $hasPart$. We assume that the TBox Σ is extended with the axiom

$$hasPart \sqsubseteq precedes$$

The role $hasPart$ represents a domain-specific relation whereas the role $precedes$ represents an "abstraction" of this role. With the verbalization technique in our example, there is no explicit part relation mentioned in the text. The part-of relation is expressed by corresponding associations in the text (linear precedence and local connectedness). The deep domain-specific interpretation is induced by abduction, and hence, as a result, the abduction process returns the Δ shown in Figure 3.23. Together with Figure 3.22, an interpretation ABox can be constructed (the "modality-specific" assertions for the role $precedes$ might be removed if appropriate). It is obvious that $hasPart$ might not be the only deep interpretation. In order to keep the discussion focussed we do not discuss further possibilities, but we keep in mind that possible alternatives might be ruled out later on due to results in fusion (see Section 3.5). In addition, it should be mentioned that even in shallow text interpretation, for instance, the tense of detected verbs could be taken into consideration and so a more liguistic-based precedence relation could be established. The example we discussed here illustrates the general principles, however.

$$
\begin{aligned}
object_1 &: High\_Jump \\
(object_1, object_2) &: hasPart \\
(object_1, object_3) &: hasPart
\end{aligned}
$$

Fig. 3.23. Addendum to the interpretation ABox shown in Figure 3.22.

3.4 Image sequence interpretation

In contrast to still images, events in image sequences have a temporal extension that has to be appropriately considered for constructing media interpretations. In order to detect high-level events such as "high-jump", event predicates are described using rules with time variables. For high-jump events we sketch the rule design pattern in Figure 3.24. In our approach we suppose that basic events can be detected by image analysis processes. Basic events are described with temporal propositions (being added to an interpretation ABox by low-level processes). An example is shown in Figure 3.25.

$$
\begin{aligned}
High\_Jump\_Event_{[T_1,T_2]}(X,Y) \leftarrow\ & accelerate\_horizontally_{[T_1,T_3]}(Y), \\
& vertical\_upward\_movement_{[T_3,T_4]}(Y), \\
& turn_{[T_4,T_5]}(Y), \\
& vertical\_downward\_movement_{[T_5,T_2]}(Y). \\
& Jumper(Y), \\
& High\_Jump(X), \\
& hasPart(X,Y),
\end{aligned}
$$

Fig. 3.24. Rule with time intervals for recognizing high jump events.

In order to actually recognize events for particular individuals which satisfy restrictions w.r.t. the ontology, the query language for temporal propositions introduced in Section 2.2 is applied. An example for a query involving events and time intervals is shown below.

$$\{(X)_{[T_1,T_2]} \mid High\_Jump\_Event_{[T_1,T_2]}(X,Y)\}$$

To answer a query, two steps have to be carried out. First, an assignment α for query variables (i.e. X in the query shown above) has to be found such that the body predicate terms and atoms are satisfied. Second, the goal is to determine lower bound and upper bound values for the temporal variables (T_1, T_2 in the example) such that the temporal propositions in the query body are satisfied. The result of the example query

$$
\begin{array}{rl}
accelerate\_horizontally_{[219,224]} & (moving\_object_1) \\
vertical\_upward\_movement_{[224,226]} & (moving\_object_1) \\
turn_{[226,228]} & (moving\_object_1) \\
vertical\_downward\_movement_{[228,230]} & (moving\_object_1) \\
moving\_object_1 : & Jumper \\
event_1 : & High\_Jump \\
(event_1, moving\_object_1) : & hasPart
\end{array}
$$

Fig. 3.25. Abox assertions for basic events (temporal propositions) that are detected by image sequence analysis components. In addition, three standard assertions possibly extracted from other sources (e.g. images and text) are added.

is $(event_1)_{[(219,223),(229,230)]}$. Thus, for all $T_1 \in (219, 223)$ and $T_2 \in (229, 230)$ and for all remaining temporal variables in the body of the rule in Figure 3.24 there exist values such that all predicate terms in the body are satisfied with the assignment $\alpha(X) \rightarrow event_1$.

If a high-jump event is expected but the query for the high-jump event (see above) returns $false$, then abduction can be used to determine what has to be added to the interpretation ABox. For instance, the temporal proposition

$$accelerate\_horizontally_{[219,224]}(moving\_object_1)$$

might probably be missing, and will be added by abduction such that the answer will be $true$ and the high jump event is "explained". It might also be the case that the image sequence analysis determined a mutilated partial basic event such as

$$accelerate\_horizontally_{[219,223]}(moving\_object_1)$$

instead. In this case, abduction would just add the proposition as in the case before. However, in this case we prefer that a near-miss is recognized, and believe that a "repair" operation for the assertion in the analysis ABox should be proposed.

3.5 Formalization of fusion

In the preceding subsections we have discussed how an interpretation ABox can be constructed for different modalities. The main idea of the approach is to use abduction and a decision procedure for determining which assertions of the analysis ABox computed by low-level analysis processes have to be explained. We did not investigate the latter decision procedure in this chapter, however.

In Figure 3.26 three interpretation pipelines for the modalities "image", "text", and "video" are shown. (see also Figure 3.3 details of the "processing pipelines" for "audio" and "image"). Let us assume that interpretation ABoxes have been computed in the "Interpretation" phase (see Figure 3.26). Actually, for every modality there might be multiple interpretation ABoxes representing multiple possibilities for high-level interpretations.

One of the problems to be solved if information from different modalities has to be combined is the *identification problem*, which is the problem of determining equality

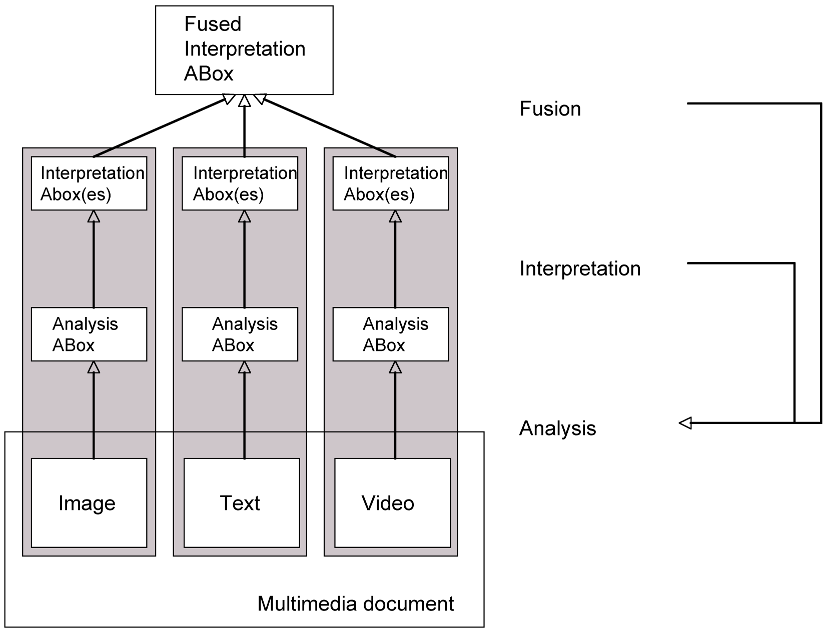

Fig. 3.26. Multimedia interpretation architecture.

assertions in order to declare co-references of different identifiers (individuals in an ABox) to the same domain objects. This problem is also relevant for single modalities (see e.g. (Gabsdil, Koller and Striegnitz 2001) for the text modality) but obviously is particularly important for multiple modalities. Heuristics, such as having the same direct types, will lead a fusion process to generate assumptions for individual equality assertions. The overall goal is to minimize the number of different domain objects (principle of Occam's Razor).

In the interpretation ABoxes for different modalities, individuals are mentioned that might refer to the same domain object. For instance, in the multimedia document about high-jump events which we used for the discussion above, there may be an image from which hj_2 and $human_2$ are extracted (see Figure 3.17). Let us assume, the image has a caption which gives rise to $object_3$ (see Figure 3.22). In addition, there could be a video from which an individual $event_1$ is extracted (see Figure 3.25). In this example, there are three different interpretation ABoxes (see Figure 3.26). Fusing these ABoxes means to construct a combined interpretation ABox. We assume that during this process, the following questions arise. Could it be the case that hj_2 and $event_1$ are names for the same event? In addition, is it reasonable to assume that $human_2$ and $object_3$ are identical? In order to test whether these assumptions do not lead to an inconsistency, the following assertions are added to the ABox.

$$hj_2 = event_1$$

$$human_2 = object_3$$

In both events the same jumper must be involved because, due to the TBox (see Figure 3.10), at most one $Jumper$ must participate in a $Jumping\_Event$ (a parent of $High\_Jump$). The resulting ABox is consistent (the unique name assumption is not applied, see Section 2.1). However, from a logical point of view, adding the above-mentioned equality assertions is not really motivated. The resulting ABox stays consistent but why should an agent assume object identity in this case? In the same spirit as we argued above, there must be a motivation for adding assertions (in the sense of assumptions). In the abduction example for constructing interpretation ABoxes that we have discussed above, adding assertions allows the agent to prove certain entailments (assumptions serve as explanations for the Γ_2 assertions). In other words, queries are answered with $true$. We believe that similar mechanisms are required for a formalization of the fusion process. The key insight is that fusing objects will allow the agent to answer certain queries, too. Consider the following example.

$$\begin{aligned} HJ\_Occurs_{[T_1,T_2]}(X) \leftarrow\ & High\_Jump(X), \\ & HJ\_InJumpPhase(X), \\ & hasPart(X,Y), \\ & Jumper(Y), \\ & vertical\_upward\_movement_{[T_1,T_3]}(Y), \\ & turn_{[T_3,T_4]}(Y), \\ & vertical\_downward\_movement_{[T_4,T_2]}(Y). \end{aligned}$$

Under the assumption that hj_2 and $event_1$ denote the same domain object, we can query the knowledge base about temporal information about a high-jump event with a famous athlete.

$$Q_3 := \{(X)_{[T_1,T_2]} \mid HJ\_Occurs_{[T_1,T_2]}(X), hasPart(X,Y), Famous(Y)\}$$

The result $(hj_2)_{[(224,225),(229,230)]}$ cannot be derived without fusing the information from multiple modalities. The query result would have been the empty set if it was not possible to prove that for $event_1$ the predicate $HJ\_InJumpPhase$ holds. This is possible due to the equality assertion $hj_2 = event_1$ (see above). Thus, fusion is motivated in this case.

Up to now it is unclear how to formalize what kinds of queries are important in a certain situation. In other words, we do not formalize what questions to ask and assume that this is represented by "context knowledge" as indicated in Figure 3.3. Context knowledge is induced by a feedback-loop from higher-level processes, which are not investigated in this work. However, feedback can also occur between analysis and interpretation. This is discussed in the next section.

3.6 Relating analysis and interpretation

Looking at the image shown in Figure 3.8 and the corresponding analysis ABox given in Figure 3.9, it becomes clear that the foam mat is not detected in this example. Even in the interpretation ABox (Figure 3.14) there is no explicit name for a foam mat involved in the pole-vault event. However, due to the TBox underlying the interpretation process (see Figure 3.10), a foam mat must exist implicitly. In other words, in all models of the interpretation ABox, the pole vault individual pv_1 is associated with a foam-mat object. If this is made explicit, feedback might be given to the analysis module (see Figure 3.26), which might use specifically parameterized image analysis techniques to then localize a foam mat in the image. In general, the more objects are made explicit in the analysis and interpretation ABoxes, the better is the interpretation. Let us assume that in the example an assertion $f_1 : Foam\_Mat$ is added to the analysis ABox, maybe together with spatial relations to the other objects localized in the scene. Then, the interpretation process will reuse the previously generated pole-vault object pv_1 and associates it with f_1 appropriately such that the following is added:

$$\begin{aligned} f_1 &: Foam\_Mat \\ (pv_1, f_1) &: hasPart \end{aligned}$$

The derivation of a complete (fused) interpretation ABox can be seen as a bootstrap process. In case the foam mat is not visible in the image, the interpretation might be considered as less plausible for a high-jump event. With the help of the distinction between domain and pictures objects and a theory for dealing with uncertainty, this is formalised in the next section.

4 Uncertain and ambiguous interpretations

Interpretations are generally ambiguous and not clearly defined with respect to a task. When constructing an explanation for media data, one often has the choice between alternatives. For example, given the limited knowledge base in Figure 3.10, the image in Figure 3.15 can be interpreted both as $Pole\_Vault$ or $High\_Jump$. In the course of a stepwise interpretation, there can be many more decision points where multiple choices are available. For example, a $High\_Jump$ or $Pole\_Vault$ may be part of a $Training\_Event$ or $Sports\_Event$. As humans, we seem to exploit our experiences for such decisions and prefer the most likely choice given all we know about the domain and the current scenario. Hence it appears natural to provide a probabilistic model for the uncertainty of logically ambiguous choices. In this section we sketch a probabilistic model which is intended to guide choices in the logic-based interpretation process presented so far.

4.1 Towards a probabilistic preference measure

The task of the probabilistic model is illustrated in Figure 3.27. In this figure we distinguish between the concepts describing real-world objects and concepts describing the corresponding media objects, a distinction which we omitted so far to simplify the presentation. All media-object concept names are marked with the suffix "pict" and describe the properties of pictures taken from the corresponding real-world objects. This way it can be explained, for example, that a real-world pole vault requires a pole but that a picture of a pole vault may not show a pole.

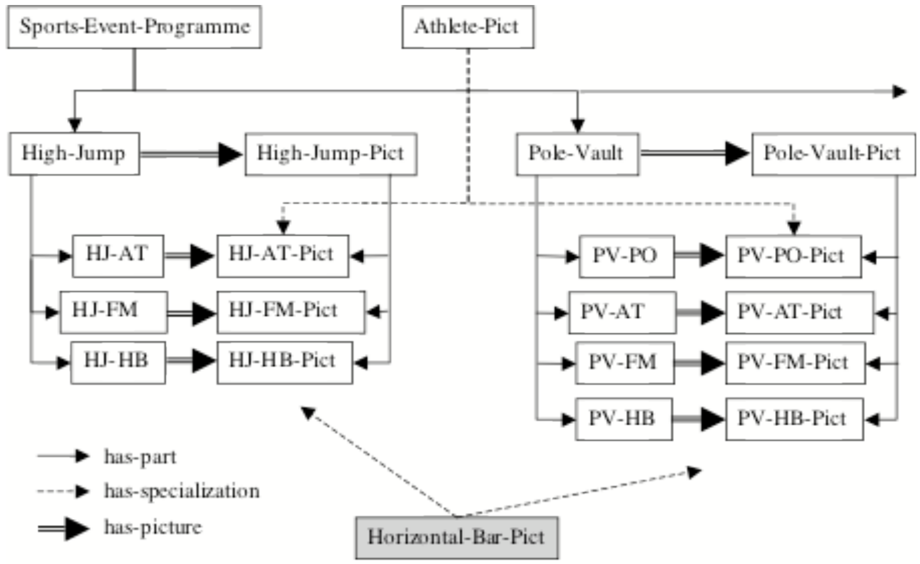

Fig. 3.27. Aggregate concepts relating a high jump and a pole vault to corresponding media object concepts. The $Horizontal{-}Bar{-}Pict$ can be interpreted as an instance of a high-jump-horizontal-bar picture ($HJ{-}HB{-}Pict$) or of a pole-vault-horizontal-bar picture ($PV{-}HB{-}Pict$).

Figure 3.27 illustrates the interpretation step where the media object $Horizontal{-}Bar{-}Pict$ must be explained. $Pole\_Vault$ and $High\_Jump$ are both logically possible, hence this is a point where a probabilistic preference measure should help.

The basic idea is to provide an estimate of how likely $Horizontal{-}Bar{-}Pict$ is a high-jump picture (i.e. an instance of $HJ{-}HB{-}Pict$) or a pole-vault picture (i.e. an instance of $PV{-}HB{-}Pict$). For this we need probability distributions such that the probabilities of one or another aggregate having a media object as part can be compared and the most probable choice can be made.

We take a frequentist approach and want the probabilities to reflect the statistics of the domain, including the statistics of corresponding media objects. Determining these statistics is, of course, a formidable task. But the example illustrates that estimates of the frequency of occurrence of pole-vault pictures without pole as opposed to the frequeny of occurrence of high-jump pictures may very well tip the balance for one interpretation rather than the other.

To compute such estimates we invoke Bayes Net technology. We consider concepts as random variables with probability distributions which govern the likelihood of instantiations which satisfy the concept. A general approach to constructing Bayes Nets for first-order logic expressions is presented in (Russell and Norvig 2003, p. 519ff.). For details see also (Koller and Pfeffer 1997; Koller and Pfeffer 1998; Pfeffer, Koller, Milch and Takusagawa 1999). Our approach exploits the fact that aggregates are the concepts of interest for an interpretation task and dependencies between objects can effectively be encapsulated in aggregates. This limits probabilistic dependencies and provides for efficient propagation mechanisms.

To show this, consider a probabilistic model for the interpretation task in Figure 3.27. We propose that each aggregate is described by a structure shown in Figure 3.28.

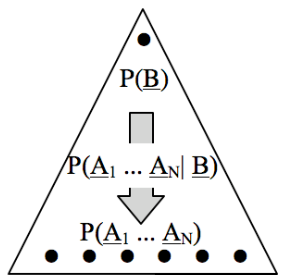

Fig. 3.28. Probabilistic structure of an aggregate. Internal properties $\underline{A}_i$ represent parts, external properties $\underline{B}$ represent the aggregate as a whole.

In the aggregate structure, we distinguish between an internal probabilistic description $P(\underline{A}_1, \underline{A}_2, ..., \underline{A}_N)$ of the parts (each represented by properties $\underline{A}_i$ and depicted as a node at the bottom of Figure 3.28) and an external, abstracted description $P(\underline{B})$ which is used to represent the aggregate as part of higher-level aggregates in the compositional hierarchy. $P(\underline{A}_1, \underline{A}_2, ..., \underline{A}_N)$ follows from $P(\underline{B})$ by means of the conditional probability distribution $P(\underline{A}_1 \ldots \underline{A}_N|\underline{B})$ which specifies the internal probabilistic structure of the aggregate.

For the time being, we consider all constituents of a high-jump event, including the pictures taken thereof, as parts of an aggregate $High{-}Jump$ and do not go into details about the internal dependency structure between real-world concepts and pictures thereof. Similarly, the pole-vault event is modelled as an aggregate $Pole{-}Vault$, and both are parts of a higher-level aggregate $Sports{-}Event{-}Programme$. The structure of an aggregate hierarchy induced by this aggregate structure is shown in Figure 3.29.

In order to provide a preference measure, we must be able to compute the effect of evidence for one node on the probabilities of other nodes. It is not obvious under which conditions this can be done based on the joint probability distributions (JPDs) of the individual aggregates as specified in Figure 3.29. The following requirements ensure that the compositional hierarchy constitutes an abstraction hierarchy where a complete JPD encompassing all aggregates can in principle be computed from the individual JPDs:

Let X be any node, $parts(X) = Y_1 \ldots Y_N$ its parts and $succ(X)$ all its successor nodes in the aggregate hierarchy. Then for a compositional hierarchy to be an abstraction hierarchy we require that

$$P(\underline{X}|succ(\underline{X})) = P(\underline{X} \mid \underline{Y}_1 \ldots \underline{Y}_N) \quad (1)$$

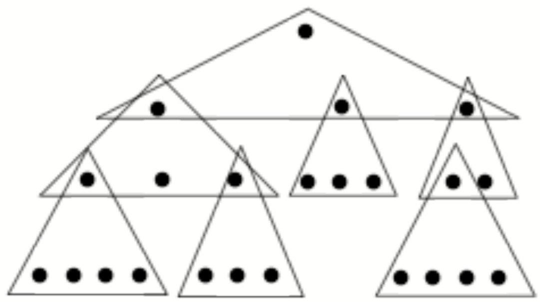

Fig. 3.29. Structure of aggregate hierarchy induced by aggregate structure. The quasi-tree structure reflects abstraction properties and allows for efficient probabilistic inferences.

Aggregate properties do not depend on details below the part properties.

$$P(succ(\underline{Y}_i) \mid \underline{Y}_1 \ldots \underline{Y}_N) = P(succ(\underline{Y}_i) \mid \underline{Y}_i) \tag{2}$$

Part properties only depend on the properties of the corresponding mother aggregate, not on correlations the mother aggregate may have as a part in a higher-level aggregate.

$$P(succ(\underline{Y}_1 \ldots \underline{Y}_N) \mid \underline{Y}_1 \ldots \underline{Y}_N) = \prod_{i=1}^{N} P(succ(\underline{Y}_i) \mid \underline{Y}_1 \ldots \underline{Y}_N) \tag{3}$$

Parts of different aggregates are statistically independent given their mother aggregates.

From (2) and (3) it follows that

$$P(succ(\underline{Y}_1 \ldots \underline{Y}_N) \mid \underline{Y}_1 \ldots \underline{Y}_N) = \prod_{i=1}^{N} P(succ(\underline{Y}_i) \mid \underline{Y}_i) \tag{4}$$

These requirements agree well with intuitions of an abstraction hierarchy. If they are fulfilled, one can show that the JPD of the aggregate hierarchy can be written as

$$P(\underline{Z}_0 \ldots \underline{Z}_M) = P(\underline{Z}_0) \prod_{i=1}^{M} P(parts(\underline{Z}_i) \mid \underline{Z}_i) \tag{5}$$

where $Z_i, i = 0 \ldots M$ are all nodes of the hierarchy and Z_0 is the root node (representing the general aggregate “any scene”).

Hence for an abstraction hierarchy, the JPD of the complete hierarchy is defined by the product of all conditional aggregate JPDs similar to a Bayes Net. Furthermore, Equation 5 applies also to branches of the abstraction hierarchy. Hence, probabilities within a branch can be compared without considering the rest of the hierarchy. For example, in the interpretation task shown in Figure 3.27, only the probabilities below the node Sports-Events-Programme have to be evaluated for choosing the most probable interpretation of Horizontal-Bar-Pict. After making the choice, Horizontal-Bar-Pict is entered into the probabilistic structure as evidence and other probabilities must be updated. Again, this can be restricted to the relevant branch of the compositional hierarchy. Updating can be performed by propagation procedures similar to those in tree-shaped Bayes Nets. On the other hand, for updates within an aggregate no simplifying dependency structure can be assumed in general, and a Bayes Net representing the internal probability structure need not be tree-shaped. This higher complexity remains local, however, due to the abstraction property.

The approach for exploiting conditional independencies discussed above allows for the construction of Bayesian networks based on the aggregate structure of domain objects. The purpose is to provide a preference measure for multiple interpretations arising naturally from abduction, and for the interpretation steps leading to such interpretations. Different from several other marriages between probabilities and logics (e.g. as discussed in the following section), our approach does not require a reinterpretation of description logic formulas but fills the space left open by multiple abduction solutions, ambiguous object classifications, and qualitative predicates over quantitative values. In consequence, there is no conceptual conflict in combining this probabilistic preference measure with a description logic framework. Combinations of description logics with approaches for modelling uncertainty are investigated in the next section.

4.2 Related work about uncertainty and description logics

Modelling uncertainty in the context of description logics has been a topic of research for many years. An overview of such extensions to classical description logics is presented in (Baader, Küsters and Wolter 2003). The research is oriented to the work of modelling uncertain knowledge on the basis of first-order structures (Nilsson 1986; Bacchus 1990; Halpern 1990). The fundamental view of the approaches based on description logics is such that it should also be possible to represent the degree of overlap between concepts (and not only subsumption or disjunction) through probabilities. Furthermore it should also be possible to formulate uncertainty about the structure of objects. Initial approaches considered primarily probabilistic knowledge at the conceptual level, this means, at the level of the TBox (Heinsohn 1994). Also knowledge representation for single objects and their relations from a probabilistic view were studied (Jaeger 1994), such that structural uncertainty could potentially be modeled. Along with early research results about decidability of very expressive logics (e.g. OWL DL), proposals for the modelling of uncertain knowledge were given.

In (Giugno and Lukasiewicz 2002), a probabilistic description logic language was studied, in which it is possible to formulate in addition to probabilistic knowledge at the conceptual level (i.e. TBox), also assertional probabilistic knowledge (i.e. ABbox) about concepts and role instances. In this language (P-$\mathcal{SHOQ}$) there is no longer a separation between TBox and ABox for the modelling of uncertainty. Its underlying reasoning formalism is based on probabilistic lexicographic entailment by (Lehmann 1995). Lexicographic entailment is based on default logic and makes use of model creation to look for preferred minimal models, where the minimal verifying (resp. falsifying) model determines entailment (resp. non-entailment). In (Giugno and Lukasiewicz 2002) the work of (Lehmann 1995) is extended from a propositional logic to a first-order logic, furthermore (Giugno and Lukasiewicz 2002) generalises classical interpretations to probabilistic interpretations by adding a probability distribution over the abstract domain and by interpreting defaults as statements of high conditional probability. E.g. in (Lehmann 1995) a default like $P(bird \rightarrow fly) \geq 1 - \varepsilon$ is in (Giugno and Lukasiewicz 2002) a conditional constraint like $l \leq P(fly|bird) \leq u$. The work of (Giugno and Lukasiewicz 2002) allows represention of probabilistic knowledge in a description logic language with high expressivity.

It is important to observe that the semantics used in the different approaches do not differ much (for example w.r.t. (Jaeger 1994) and (Giugno and Lukasiewicz 2002)). An approach for the modelling of uncertain structures for a less expressive language is presented in (Dürig and Studer 2005). However, no specific inference algorithms are known for this approach. An important step for the practical use of description logics with probabilities occurred with the integration of Bayesian networks in P-CLASSIC (Koller, Levy and Pfeffer 1997), nevertheless very strong disadvantages were obtained: for number restrictions the supremum limits must be known and separate Bayesian networks are necessary to consider role fillers. Along with this problem, the probabilistic dependencies between instances must also be modeled. This problem was overcome in (Koller and Pfeffer 1998) - however not in the context of description logics but with a frame-based approach, in which the treatment of default values is given without formal semantics. The main idea in (Koller and Pfeffer 1998) is the view of considering role fillers as nodes in Bayesian networks which have CPTs (conditional probability tables) associated to them as generalized number restrictions in the sense of description logics. Related studies followed in (Pfeffer et al. 1999).

Complementary to the P-CLASSIC approach, another approach called PTDL (Yelland 2000) was developed for probabilistic modelling with the use of first-order structures. In this approach the Bayesian network theory is considered as basis reference for further extensions, instead of (classical) description logics. The Bayesian network nodes represent function values and an individual is associated to other nodes through these function values. The approach in (Yelland 2000) avoids some disadvantages of P-CLASSIC, but it offers minimal expressivity on the side of description logics. In context with very expressive description logics another approach (Ding and Peng 2004; Ding, Peng and Pan 2005) was presented for the integration of Bayes networks. Algorithms for deduction over probabilistic first-order structures were developed by Poole (Poole 2003). Poole observes, that the existing approaches (e.g. (Koller and Pfeffer 1998; Pfeffer et al. 1999)) only consider individuals that are explicitly named. Qualitative probabilistic matching with hierarchical descriptions was studied (Smyth and Poole 2004). It allows for a variation of the level of abstraction.

Previous studies have investigated the combination of Datalog and description logics (so-called description logic programs) (Nottelmann and Fuhr 2004; Lukasiewicz 2005*a*; Lukasiewicz 2005*b*; Nottelmann and Fuhr 2006). Approaches for information retrieval with probabilistic Datalog are presented in (Fuhr 2000; Fuhr 1995). In this area, work on learning from Datalog-predicates with uncertainty is also relevant (Nottelmann and Fuhr 2001).

Modelling vagueness to capture notions of imprecise knowledge has been intensively studied (Straccia 2001; Tresp and Molitor 1998; Yen 1991), such that existing knowledge representation formalisms like first-order logic can be extended to represent vague concepts (e.g. hot, cold) which are not entirely true or false, but rather have a truth value between true and false. Fuzzy Logic, with a basis in fuzzy set theory, allows the modelling of vagueness, and its fundamental view is that the classical ideas of satisfiability and subsumption are modified such that concepts are satisfiable to a certain degree, or a concept subsumes another to a certain degree.

In (Tresp and Molitor 1998) a tableau-like method for computing the degree of subsumption between two concepts in the language $\mathcal{ALC}_{fm}$ was presented. In (Yen 1991) work on extending description logics with fuzzy features is presented for the language FL$^-$, in which it is possible to determine subsumption, but not possible to determine whether an individual is an instance of a concept with a certain probability. In (Straccia 2001), the use of fuzzy logic is highlighted in the context of multimedia information retrieval, in which images are semantically annotated with fuzzy statements. Recently, more expressive fuzzy description logics have been investigated (Stoilos, Stamou, Tzouvaras, Pan and Horrocks 2005*b*; Stoilos, Stamou, Tzouvaras, Pan and Horrocks 2005*c*; Stoilos, Stamou, Tzouvaras, Pan and Horrocks 2005*a*; Pan, Stoilos, Stamou, Tzouvaras and Horrocks 2006; Stoilos, Straccia, Stamou and Pan 2006; Stoilos, Stamou and Pan 2006).

At the time of this writing, details of a probabilistic (and fuzzy) inference scheme for media interpretation in a local context are still being investigated in ongoing research. One of the open questions is how to trade off precision (which may not be vital for a preference measure) against computational effort (which may be unaccepable if all dependencies in a large knowledge base have to be considered).

4.3 Probabilities, description logics, abduction, and logic programming

While (Hobbs et al. 1993; Shanahan 2005) use first-order logic for text and image/video interpretation, with description logics, we use a decidable knowledge representation formalism with well-tested implementations that are known to be efficient for many typical-case inputs. The use of logical rules and backward chaining for implementing an abduction algorithm as described in Section 3 is also investigated in the area of logic programming (Kakas, Kowalski and Toni 1992; Poole 1993*a*; Poole 1992; Kakas and Denecker 2002; Flach and Kakas 2000). In our approach, however, predicate names in rules are defined w.r.t. ontologies represented as description logic Tboxes, and thus we use another expressive fragment of first-order logic. In the context of information retrieval, user queries can be answered regarding user-specified Tboxes. In the previous sections, we have argued that probabilistic reasoning would really add to the application scenario of information retrieval we have used in this chapter. In (Sebastiani 1994) an proposal is made for using probabilistic description logics for information retrieval. No system implementation has been developed, though.

In the previous section we have discussed related work for integrating probabilistic and description logic reasoning. Only recently, however, abduction has been investigated in the context of description logics (Colucci, Noia, Sciascio, Mongiello and Donini 2004). However, in this work, abduction is considered for concepts, not Aboxes and queries. Due to the best of our knowledge, abduction has not yet been considered in the context of probabilistic description logics. Interesting input to this research is provided by abduction in probabilistic logic programming (Charniak and Goldman 1991; Poole 1993*b*).

5 Conclusions

In this chapter a formal account of media interpretation has been presented. It has been shown that results from media analysis processes can be appropriately enriched with high-level descriptions using automatic processes. Thus, applications which require access to high-level descriptions can be supported, for example, retrieval of media in a Semantic Web context. The central idea is to use abduction and deduction in concert to construct high-level descriptions for characterizing media content. It should be emphasise that the architecture not just describes an algorithm that constructs the descriptions but formalises the description generation in a meaningful way. High-level descriptions are constructed to explain assertions from media analysis. The same holds also for the way we tackle the fusion problem. In its current state, our architecture does not specify which assertions are taken to be explained (and which queries are constructed or selected in the case of fusion). This is seen as an even higher-level process and is left for future work

It has also been shown that the crisp logical framework should be supported by probabilistic preference measures in order to provide the most desirable interpretations. We have made a first contribution towards a probabilistic preference measure which can be used to rank different interpretations.

Acknowledgements

This work is partially supported by the EU-funded projects BOEMIE (Bootstrapping Ontology Evolution with Multimedia Information Extraction, IST-FP6-027538) and TONES (Thinking ONtologiES, IST-FP6-007603). We would like to thank all members of the BOEMIE project for providing examples about the sports domains that we use here to demonstrate insights about media interpretation and reasoning. In particular, we would like to mention Sofia Espinosa, Atila Kaya, and Sylvia Melzer.

References

Baader, F., Calvanese, D., McGuinness, D., Nardi, D. and Patel-Schneider, P. F., eds (2003), *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press.

Baader, F., Franconi, E., Hollunder, B., Nebel, B. and Profitlich, H.-J. (1994), 'An empirical analysis of optimization techniques for terminological representation systems or: Making KRIS get a move on', *Applied Artificial Intelligence. Special Issue on Knowledge Base Management* **4**, 109–132.

Baader, F. and Hanschke, P. (1991), A Schema for Integrating Concrete Domains into Concept Languages, *in* 'Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)', pp. 452–457.

Baader, F., Küsters, R. and Wolter, F. (2003), Extensions to description logics, *in* Baader, Calvanese, McGuinness, Nardi and Patel-Schneider (2003), chapter 6, pp. 219–261.

Baader, F., Lutz, C. and Suntisrivaraporn, B. (2006), CEL—a polynomial-time reasoner for life science ontologies, *in* U. Furbach and N. Shankar, eds, 'Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06)', Vol. 4130 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 287–291.

Bacchus, F. (1990), *Representing and reasoning with probabilistic knowledge: A logical approach to probabilities*, The MIT Press, Cambridge.

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M. and Rosati, R. (2005), DL-Lite: Tractable description logics for ontologies, *in* 'Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)', pp. 602–607.

Charniak, E. and Goldman, R. (1991), Probabilistic abduction for plan recognition, Technical report, Brown University, Providence, RI, USA.

Colucci, S., Noia, T. D., Sciascio, E. D., Mongiello, M. and Donini, F. M. (2004), Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace, *in* 'ICEC '04: Proceedings of the 6th international conference on Electronic commerce', ACM Press, New York, NY, USA, pp. 41–50.

Di Sciascio, E., Donini, F. M. and Mongiello, M. (1999), A description logic for image retrieval, *in* 'Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence', number 1792 *in* 'Lecture Notes in Computer Science', Springer, pp. 13–24.

Di Sciascio, E., Donini, F. and Mongiello, M. (2000), Semantic indexing in image retrieval using description logic, *in* 'Proceedings of the 22nd International Conference on Information Technology Interfaces (ITI 2000)', pp. 125–132.

Ding, Z. and Peng, Y. (2004), A probabilistic extension to ontology language OWL, *in* 'Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS)'.

Ding, Z., Peng, Y. and Pan, R. (2005), BayesOWL: Uncertainty modeling in semantic web ontologies, *in* 'Soft Computing in Ontologies and Semantic Web', Springer.

Dürig, M. and Studer, T. (2005), Probabilistic abox reasoning: Preliminary results, *in* 'Proc. Int. Description Logics Workshop 2005', pp. 104–111.

Elsenbroich, C., Kutz, O. and Sattler, U. (2006), A case for abductive reasoning over ontologies, *in* 'Proc. OWL: Experiences and Directions, Athens, Georgia, USA, November 10-11'.

Erman, L. D., Hayes-Roth, F., Lesser, V. R. and Reddy, D. R. (1980), 'The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty', *ACM Computing Surveys* **12**(2), 213–253.

Flach, P. and Kakas, A., eds (2000), *Abduction and Induction: Essays on their relation and integration*, Kluwer Academic Publishers.

Fuhr, N. (1995), Probabilistic Datalog: a logic for powerful retrieval methods, *in* 'Proceedings of SIGIR-95: 18th ACM International Conference on Research and Development in Information Retrieval', pp. 282–290.

Fuhr, N. (2000), 'Probabilistic datalog: Implementing logical information retrieval for advanced applications', *Journal of the American Society of Information Science* **51**(2), 95–110.

Gabsdil, M., Koller, A. and Striegnitz, K. (2001), Building a text adventure on description logic, *in* 'International Workshop on Applications of Description Logics', CEUR Electronic Workshop Proceedings.

Giugno, R. and Lukasiewicz, T. (2002), P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web, *in* 'JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence', Springer-Verlag, pp. 86–97.

Glimm, B., Horrocks, I., Lutz, C. and Sattler, U. (2007), Conjunctive query answering for the description logic $\mathcal{SHIQ}$, *in* 'Proceedings of the Twentieth International Joint Conference on Artificial Intelligence IJCAI-07', AAAI Press.

Haarslev, V. and Möller, R. (2000), Expressive ABox reasoning with number restrictions, role hierarchies, and transitively closed roles, *in* 'Proc. of the 7th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2000)', pp. 273–284.

Haarslev, V. and Möller, R. (2001), RACER System Description, *in* 'Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001)', Vol. 2083 of *Lecture Notes in Computer Science*, Springer, pp. 701–705.

Haarslev, V., Möller, R. and Wessel, M. (2001), The description logic $\mathcal{ALCNH}_{R+}$ extended with concrete domains: A practically motivated approach, *in* 'Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001)', pp. 29–44.

Halpern, J. (1990), 'An analysis fo first-order logics of probability', *Artificial Intelligence* **46**(3), 311–350.

Hanson, A. and Riseman, E. (1978), VISIONS: A computer system for interpreting scenes, *in* A. Hanson and E. Riseman, eds, 'Computer Vision Systems', Academic Press, New York, pp. 303–333.

Heinsohn, J. (1994), Probabilistic description logics, *in* R. L. de Mantaras and D. Poole, eds, 'Proc. of the 10th Conf. on Uncertainty in Artificial Intelligence', Morgan Kaufmann, Seattle, Washington, pp. 311–318.

Hobbs, J. R., Stickel, M., Appelt, D. and Martin, P. (1993), 'Interpretation as abduction', *Artificial Intelligence* **63**, 69–142.

Hobbs, J. R., Stickel, M. E., Appelt, D. and Martin, P. (1990), Interpretation as abduction, Technical Report Technical Report 499, AI Center, SRI International, Menlo Park, California.

Hobbs, J. R., Stickel, M., Martin, P. and Edwards, D. D. (1988), Interpretation as abduction, *in* '26th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference', Buffalo, New York, pp. 95–103.

Horrocks, I., Kutz, O. and Sattler, U. (2006), The even more irresistible $\mathcal{SROIQ}$, *in* 'Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)', AAAI Press, pp. 57–67.

Horrocks, I., Sattler, U., Tessaris, S. and Tobies, S. (2000), How to decide query containment under constraints using a description logic, *in* 'Proc. of the 7th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR 2000)', Vol. 1955 of *Lecture Notes in Computer Science*, Springer, pp. 326–343.

Horrocks, I., Sattler, U. and Tobies, S. (2000), Reasoning with individuals for the description logic $\mathcal{SHIQ}$, *in* D. McAllester, ed., 'Proc. of the 17th Int. Conf. on Automated Deduction (CADE 2000)', Vol. 1831 of *Lecture Notes in Computer Science*, Springer, pp. 482–496.

Hustadt, U., Motik, B. and Sattler, U. (2004), Reducing $\mathcal{SHIQ}$-Description Logic to Disjunctive Datalog Programs, *in* 'Proc. of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2004)', pp. 152–162.

Jaeger, M. (1994), Probabilistic reasoning in terminological logics, *in* 'Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'94)', pp. 305–316.

Kakas, A. C., Kowalski, R. A. and Toni, F. (1992), 'Abductive logic programming', *Journal of Logic and Computation* **2**(6), 719–770.

Kakas, A. and Denecker, M. (2002), Abduction in logic programming, *in* A. Kakas and F. Sadri, eds, 'Computational Logic: Logic Programming and Beyond. Part I', number 2407 *in*

'LNAI', Springer, pp. 402–436.

Kaplunova, A., Kaya, A. and Möller, R. (2006), First experiences with load balancing and caching for semantic web applications, Technical report, Institute for Software Systems (STS), Hamburg University of Technology, Germany.

Koller, D., Levy, A. and Pfeffer, A. (1997), P-CLASSIC: A tractable probabilistic description logic, *in* 'Proc. of the 14th Nat. Conf. on Artificial Intelligence (AAAI'97)', AAAI Press/The MIT Press, pp. 390–397.

Koller, D. and Pfeffer, A. (1997), Object-oriented Bayesian networks, *in* 'Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI)', pp. 302–313.

Winner of the Best Student Paper Award.

Koller, D. and Pfeffer, A. (1998), Probabilistic frame-based systems, *in* 'Proceedings of the 15th National Conference on Artificial Intelligence (AAAI), Madison, Wisconsin'.

Lehmann, D. J. (1995), 'Another perspective on default reasoning', *Annals of Mathematics and Artificial Intelligence* **15**(1), 61–82.

Lukasiewicz, T. (2005*a*), Probabilistic description logic programs, *in* 'Proc. of ECSQARU', pp. 737–749.

Lukasiewicz, T. (2005*b*), Stratified probabilistic description logic programs, *in* 'Proc. of ISWC-URSW', pp. 87–97.

Möller, R., Haarslev, V. and Neumann, B. (1998), Semantics-based information retrieval, *in* 'Proc. IT&KNOWS-98: International Conference on Information Technology and Knowledge Systems, 31. August- 4. September, Vienna, Budapest', pp. 49–62.

Motik, B., Sattler, U. and Studer, R. (2005), 'Query answering for OWL-DL with rules', *J. of Web Semantics* **3**(1), 41–60.

Neumann, B. (1985), Retrieving events from geometrical descriptions of time-varying scenes, *in* J. Schmidt and C. Thanos, eds, 'Foundations of Knowledge Base Management – Contributions from Logic, Databases, and Artificial Intelligence', Springer, p. 443.

Neumann, B. and Möller, R. (2006), On scene interpretation with description logics, *in* H. Christensen and H.-H. Nagel, eds, 'Cognitive Vision Systems: Samping the Spectrum of Approaches', number 3948 *in* 'LNCS', Springer, pp. 247–278.

Neumann, B. and Novak, H.-J. (1983), Event models for recognition and natural language description of events in real-world image sequences, *in* 'Proc. of the 8th Int. Joint Conf. on Artificial Intelligence (IJCAI'83)', pp. 724–726.

Neumann, B. and Weiss, T. (2003), Navigating through logic-based scene models for high-level scene interpretations, *in* '3rd International Conference on Computer Vision Systems - ICVS 2003', Springer, pp. 212–22.

Nilsson, N. (1986), 'Probabilistic logic', *Artificial Intelligence* **28**, 71–87.

Nottelmann, H. and Fuhr, N. (2001), Learning probabilistic datalog rules for information classification and transformation, *in* 'In Proceedings CIKM', pp. 387–394.

Nottelmann, H. and Fuhr, N. (2004), pDAML+OIL: A probabilistic extension to DAML+OIL based on probabilistic datalog, *in* 'Proceedings Information Processing and Management of Uncertainty in Knowledge-Based Systems'.

Nottelmann, H. and Fuhr, N. (2006), 'Adding probabilities and rules to OWL Lite subsets based on probabilistic datalog', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **14**(1), 17–41.

Pan, J. Z., Stoilos, G., Stamou, G., Tzouvaras, V. and Horrocks, I. (2006), 'f-swrl: A fuzzy extension of swrl', *Data Semantics, special issue on Emergent Semantics* **4090**, 28–46.

Pfeffer, A., Koller, D., Milch, B. and Takusagawa, K. (1999), SPOOK: A system for probabilistic object-oriented knowledge representation, *in* 'Proc. of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)', pp. 541–550.

Poole, D. (1992), Logic programming, abduction and probability, *in* 'Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'92),', pp. 530–538.

Poole, D. (1993*a*), 'Logic programming, abduction and probability: a top-down anytime algorithm for estimating prior and posterior probabilities', *New Generation Computing* **11**(3-4), 377–400.

Poole, D. (1993*b*), 'Probabilistic horn abduction and bayesian networks', *AIJ* **64**(1), 81–129.

Poole, D. (2003), First-order probabilistic inference, *in* 'Proc. International Joint Conference on Artificial Intelligence IJCAI-03', pp. 985–991.

Reiter, R. and Mackworth, A. (1990), 'A logical framework for depiction and image interpretation', *Artificial Intelligence* **41**, 125–155.

Russell, S. J. and Norvig, P. (2003), *Artifical Intelligence: A Modern Approach*, Prentice Hall, 2nd edition.

Schober, J.-P., Hermes, T. and Herzog, O. (2005), Picturefinder: Description logics for semantic image retrieval, *in* 'IEEE International Conference on Multimedia and Expo (ICME 2005)'.

Schröder, C. (1999), Bildinterpretatatation durch Modellkonstruktion: Eine Theorie zur rechnergestützten Analyse von Bilder, PhD thesis, Universität Hamburg.

Sebastiani, F. (1994), A Probabilistic Terminological Logic for Modelling Information Retrieval, *in* W. Croft and C. v. Rijsbergen, eds, 'Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', Springer-Verlag, Dublin, Ireland, pp. 122–130.

Shanahan, M. (2005), 'Perception as abduction: Turning sensor data into meaningful representation', *Cognitive Science* **29**, 103–134.

Sirin, E. and Parsia, B. (2006), Pellet System Description, *in* 'Proc. of the 2006 Description Logic Workshop (DL 2006)', CEUR Electronic Workshop Proceedings.

Smyth, C. and Poole, D. (2004), Qualitative probabilistic matching with hierarchical descriptions, *in* 'Proc. Knowledge Representation and Reasoning (KR&R 2004)'.

Stoilos, G., Stamou, G. and Pan, J. (2006), Handling imprecise knowledge with fuzzy description logic, *in* 'International Workshop on Description Logics (DL 06), Lake District, UK'.

Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J. and Horrocks, I. (2005*a*), The fuzzy description logic f-shin, *in* 'International Workshop on Uncertainty Reasoning For the Semantic Web'.

Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J. and Horrocks, I. (2005*b*), A fuzzy description logic for multimedia knowledge representation, *in* 'Proc. of the International Workshop on Multimedia and the Semantic Web'.

Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J. and Horrocks, I. (2005*c*), Fuzzy owl: Uncertainty and the semantic web, *in* 'International Workshop of OWL: Experiences and Directions, Galway'.

Stoilos, G., Straccia, U., Stamou, G. and Pan, J. Z. (2006), General concept inclusions in fuzzy description logics, *in* '17th European Conference on Artificial Intelligence (ECAI 06), Riva del Garda, Italy'.

Straccia, U. (2001), 'Reasoning within fuzzy description logics', *J. of Artificial Intelligence Research* **14**, 137–166.

Tresp, C. B. and Molitor, R. (1998), A description logic for vague knowledge, *in* 'Proc. of the 13th Eur. Conf. on Artificial Intelligence (ECAI'98)', pp. 361–365.

Tsarkov, D. and Horrocks, I. (2006), FaCT++ Description Logic Reasoner: System Description, *in* 'Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)'.
To appear.

Turhan, A.-Y., Bechhofer, S., Kaplunova, A., Liebig, T., Luther, M., Möller, R., Noppens, O., Patel-Schneider, P., Suntisrivaraporn, B. and Weithöner, T. (2006), DIG 2.0 – towards a flexible interface for description logic reasoners, *in* B. C. Grau, P. Hitzler, C. Shankey and E. Wallace, eds, 'OWL: Experiences and Directions 2006'.

Wessel, M. and Möller, R. (2005), A High Performance Semantic Web Query Answering Engine, *in* I. Horrocks, U. Sattler and F. Wolter, eds, 'Proc. International Workshop on Description Logics'.

Wessel, M. and Möller, R. (2006), A flexible DL-based architecture for deductive information systems, *in* G. Sutcliffe, R. Schmidt and S. Schulz, eds, 'Proc. IJCAR-06 Workshop on Empirically Successful Computerized Reasoning (ESCoR)', pp. 92–111.

Yelland, P. Y. (2000), An alternative combination of Baysian networks and description logics, *in* 'Proc. of the 7th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2000)', pp. 225–234.

Yen, J. (1991), Generalizing term subsumption languages to fuzzy logic, *in* 'Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)', pp. 472–477.