Coordinated reasoning with inference fusion

Bo Hu[1] Ernesto Compatangelo[2] Inés Arana[1]
[1]*School of Computing, the Robert Gordon University, Aberdeen AB25 1HG, Scotland, UK*
[2]*Department of Computing Science, University of Aberdeen, AB26 1HG, Scotland, UK*

Abstract We discuss a new approach which uses inference fusion, i.e. the cooperative reasoning from distributed heterogeneous inference systems, in order to extend the scope of deductions based on description logics. More specifically, our approach integrates results from a description logics reasoner with results from a constraint solver. Inference fusion (i) processes heterogeneous input knowledge, generating suitable homogeneous input knowledge for each specialised reasoner; (ii) passes control to each reasoner, collecting their results and making them available to the other reasoner for further inferencing; (iii) combines the results of the two reasoners. We outline the main features of inference fusion by way of a small example.

1 Introduction

Inferential engines based on Description Logics (DLs) are extremely powerful when reasoning about taxonomic knowledge, since they can discover hidden subsumption relationships amongst classes. However, their expressive power is restricted in order to reduce the computational complexity and to guarantee the decidability of their deductive algorithms. Consequently, this restriction prevents taxonomic reasoning from being widely applicable to heterogeneous domains (e.g. integer and rational numbers, strings) in practice.

Several extensions of DLs, which incorporate concrete domains have been proposed in the past, e.g. $\mathcal{ALC}(\mathcal{D})$ [1]. However, these extensions are generally not implemented, with the exception of integer number restrictions in systems such as CLASSIC [2] and RACER [5]. This latter system extends the algorithm originally devised for $\mathcal{ALC}$ [9], i.e. it creates a tableaux containing both concept constructors and constraint predicates.

In this paper, we present a different kind of approach, which is based on the cooperative reasoning from distributed heterogeneous inference systems. We denote this approach, which extends DL systems with constraint reasoning without increasing their complexity, as *inference fusion*. In our approach, no additional DL constructors or operators are introduced. Instead, results from a DL engine and a constraint solver are shared and fused as appropriate.

2 A hybrid representation for heterogeneous domains

We use the $\mathcal{ALC}$ DL language as the basis for our hybrid representation since it provides most of the constructs we need. The interpretation of $\mathcal{ALC}$ constructors can be found in [9]. However, some engineering and architectural domains require different numeric constraints which further restrict roles properties. New constructs are thus necessary to express this type of knowledge. Therefore, we have introduced a new DL-based modelling language, called $\mathcal{DL}(\mathrm{D})/\mathrm{S}$ [7, 8], which extends $\mathcal{ALC}$ by incorporating the constraints shown in Table 1.

Table 1: Syntax and semantics of $\mathcal{DL}(\mathrm{D})/\mathrm{S}$ extension

Constructor	**Syntax**	**Semantics** (*Interpretation*)
role value constraint(D)	$\forall_{\mathrm{H}}\mathsf{R}.\mathsf{H}$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x,\ y\rangle \in \mathsf{R}^{\mathcal{I}} \ \rightarrow\ y \in \mathsf{H}^{\mathcal{I}}\}$
role cardinality constraint(S)	(*rel* v R) $\exists v.\mathsf{C}[v]/\psi[v]$	$\{\, c \in \Delta^{\mathcal{I}} \vert \sharp \,\{\, d \in \Delta^{\mathcal{I}} : \langle\, c, d\,\rangle \in \mathsf{R}^{\mathcal{I}} \,\}\ rel\ \lambda'(\mathbf{v})\ \}$ $\mathsf{C}^{\mathcal{I}}[\lambda(v)]$ where $\psi[\lambda(v)]$ hold

Here, $rel \in \{=\}$, H is a hybrid concept, v an integer type variable; λ' an assignment mapping v to a set of non-negative integers. $\mathcal{DL}(\mathrm{D})/\mathrm{S}$ constraints are, therefore, specified through hybrid role successors H, i.e. $\forall x_1 \in \mathsf{H}_1, \ldots, \forall x_n \in \mathsf{H}_n.\mathsf{Q}(x_1, \ldots, x_n)$ or symbolic role number restrictions (role cardinalities, RC for short), i.e. $\exists v_1, \ldots, v_n.\mathrm{P}(v_1, \ldots, v_n)$ where $\mathsf{H}_1, \ldots, \mathsf{H}_n$ are hybrid concepts, $v_1, \ldots, v_n$ role cardinality variables and Q, P are constraint predicates. For example, the following concept contains a numeric constraint which restricts the number of airpads to be twice the number of axis:

$$\exists(\alpha, \beta).(\mathsf{Machine\_Tool} \sqcap (=\ \alpha\ \mathsf{has\text{-}axis}) \sqcap (=\ \beta\ \mathsf{has\text{-}airpad}))/\{\alpha = 2\beta\}$$

3 Hybrid reasoning: inference fusion of DL-based and constraint-based inferences

In order to exploit the power of DL-based approaches when reasoning about heterogeneous domains we have developed a hybrid approach called *inference fusion* [8]. This is defined as a three-stage process (namely, knowledge splitting, homogeneous reasoning and inference combination) which facilitates the collaboration of two disparate reasoners.

Inference fusion combines TBox deductions from a taxonomic reasoner with constraint satisfaction inferences from a constraint solver (CS) under the direction of a reasoning coordinator. In order to ensure the autonomy of both reasoners, this coordinator uses *linkages*, i.e. relations mapping objects from one system to another. The architecture of our system (see Figure 1) is composed of a *reasoning coordinator*, two *engine interfaces*, a *user (and KB) interface*, the *internal storage*, a *DL reasoner* and a *constraint reasoner*. In our architecture, called CONCOR, no reasoner behaves as a sub-system of the other.

The *reasoning coordinator* is at the heart of CONCOR and is responsible for redirecting knowledge to the specialised reasoning engines, analysing the results of their inferencing, deciding whether further processes should be carried out, updating the Knowledge Base (KB) files and returning results. It uses an internal language which acts as the mediator between the input representation and the underlying representation used by the selected DL or CS system. This language (i) makes CONCOR engine-independent, since whenever a reasoner is replaced, only its interface must be developed, and (ii) reduces the programming effort on any further extensions to the modelling language, since only the parser residing in the *user (and KB) interface* needs to be upgraded. A parser is included in the *user (and KB) interface* in order to analyse and fragment the input descriptions into three sets of statements:

- A set of DL statements Π_{DL} which do not exceed the expressive power of the DL system;
- A set of non-DL statements $\Pi_{\text{non-DL}}$ which contains the concrete knowledge not in Π_{DL};
- A set of *linkages* $\Pi_{linkage}$ which are 1:1 relations connecting DL and non-DL statements.

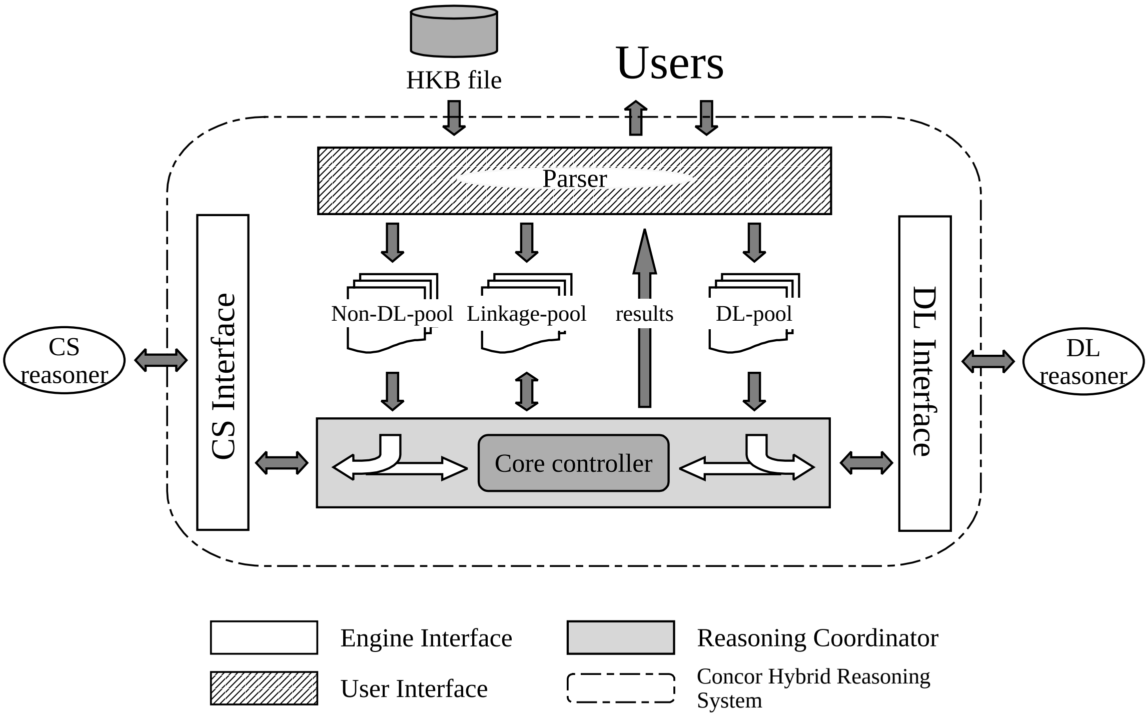

Figure 1: CONCOR's architecture

As a result, all the information related to the numeric constraints is removed from the concept definitions, leaving only proper DL constructors which are admitted by the selected DL inferential engine. Thus Π_{DL}, $\Pi_{\text{non-DL}}$, and $\Pi_{Linkage}$ are stored into the three *pools*, respectively denoted as *DL*, *non-DL*, and *linkage*, which form the *internal storage*.

Two features are beneficial for reasoning in the CONCOR architecture: (i) DL systems can specify (told) subsumption relationships between concepts, and (ii) an ordering (denoted as *quasi-ordering* [8]) can be introduced between different sets of constraints. The CONCOR architecture handles numeric constraints in two different ways: (i) global constraints are applied by reducing the domains of constrained objects (i.e. maintaining a path consistency among the objects); (ii) local constraints are enhanced by explicitly expressing the restrictions which are otherwise implicit, i.e. *quasi-ordering* and disjointness among concepts.

An example of fusion of DL-based and constraint-based inferences is shown in Figure 2, where the two concepts Midsideorg and Organisation1 (see point ❶) are defined as follows.

- Midsideorg is an organisation with an annual net income $\pounds 10,000 \leq \mathsf{X}_{ai} \leq \pounds 50,000$, and a number of female employees as a proportion of the number of male employees.
- Organisation1 is an organisation with an annual net income of $\pounds 25,000$ (i.e. a quarter of the gross income $\pounds 100,000$), 50 female employees, and 100 male employees.

Hybrid reasoning with inference fusion using the above two concepts is performed as follows.

Firstly, the hybrid KB in point ❶ (written in $\mathcal{DL}(\mathrm{D})/\mathrm{S}$) is parsed for (i) illegal syntax, and (ii) illegal constructors, i.e. those constructors that are not admitted by the selected inferential engines. The well-formed concept descriptions are normalised and translated into the intermediate language and split into DL, non-DL, and linkage pools as shown in point ❷.

❶ Hybrid concepts:

$$\begin{aligned}
\mathsf{Midsizeorg} \doteq\ & \mathsf{Organisation} \sqcap \\
& (\forall_h\ \mathsf{annual\text{-}income.}\ \mathsf{X}_{ai}) \sqcap \\
& ((=\ \alpha\ \ \mathsf{female\text{-}emp}) \sqcap (=\ \beta\ \ \mathsf{male\text{-}emp})) \\
& /\exists\alpha\exists\beta.P(\alpha,\beta)
\end{aligned}$$

$$\begin{aligned}
\mathsf{Organisation1} \doteq\ & \mathsf{Organisation} \sqcap \\
& (\forall_h\ \mathsf{annual\text{-}income.}\ \mathsf{X}_{ai1}) \sqcap \\
& ((=\ 50\ \ \mathsf{female\text{-}emp}) \sqcap (=\ 100\ \ \mathsf{male\text{-}emp}))
\end{aligned}$$

$$\begin{aligned}
& 10,000 \leq \mathsf{X}_{ai} \leq 50,000 \\
& \mathsf{X}_{ai1} = Q(X_{\text{gross-income}}) \\
& X_{\text{gross-income}} = 100,000
\end{aligned}$$

❷ Concepts are normalised as:

$$\begin{aligned}
\mathsf{Midsizeorg} \doteq\ & \mathsf{Organisation} \sqcap \\
& (\forall\ \mathsf{annual\text{-}income.}\ \mathsf{X}_{ai}) \sqcap \\
& ((\exists\ \mathsf{female\text{-}emp}) \sqcap (\exists\ \mathsf{male\text{-}emp}))\ \sqcap\ \mathsf{C}_{midsize}
\end{aligned}$$

$$\begin{aligned}
\mathsf{Organisation1} \doteq\ & \mathsf{Organisation} \sqcap \\
& (\forall\ \mathsf{annual\text{-}income.}\ \mathsf{X}_{ai1}) \sqcap \\
& ((\exists\ \mathsf{female\text{-}emp})\ \sqcap (\exists\ \mathsf{male\text{-}emp}))\ \sqcap\ \mathsf{C}_{org1}
\end{aligned}$$

$$\begin{aligned}
& 10,000 \leq \mathsf{X}_{ai} \leq 50,000 \\
& \mathsf{X}_{ai1} = Q(X_{\text{gross-income}}) \\
& X_{\text{gross-income}} = 100,000 \\
\mathsf{C}_{midsize} =\ & \exists\alpha\exists\beta. \quad (|\mathsf{female\text{-}emp}| = \alpha \wedge \\
& \qquad |\mathsf{male\text{-}emp}| = \beta \wedge P(\alpha,\beta)) \\
\mathsf{C}_{org1} =\ & (|\mathsf{female\text{-}emp}| = 50 \wedge |\mathsf{male\text{-}emp}| = 100)
\end{aligned}$$

where $X_{\text{gross-income}}$ is a constant.

❸ After constraint reasoning, *quasi-ordering* is obtained among non-DL concepts. Assuming that $\mathsf{X}_{ai1} \subseteq \mathsf{X}_{ai}$, $\mathsf{C}_{org1} \models \mathsf{C}_{midsize}$

❹ “Told” Subsumption relationships are created among hybrid concepts.

❺ Based on ❹, overall reasoning result is :

$$\mathsf{Organisation1} \sqsubseteq \mathsf{Midsizeorg}$$

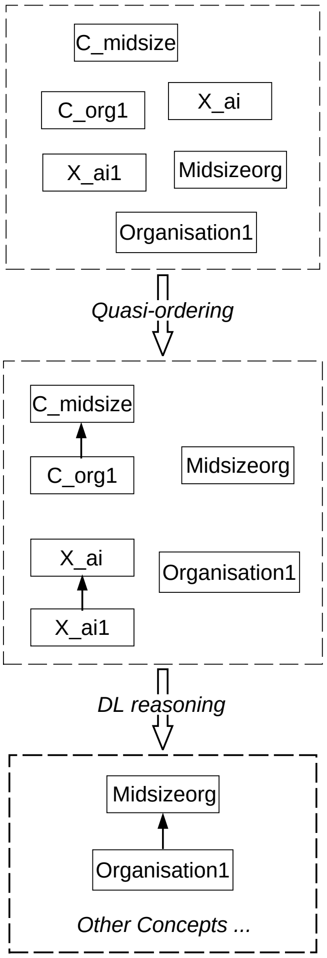

Figure 2: An example of hybrid reasoning with inference fusion in CONCOR

Secondly, the *user (and KB) interface* passes control to the *reasoning coordinator*, which decides what reasoner should be activated. In our implementation, the consistency of all global constraints is first checked using the constraint reasoner. Afterwards, the constraints are propagated and, consequently, the domains of constrained variables are reduced as shown in point ❷ (i.e. they are hierarchically linked as shown in the middle pane of Figure 2).

Thirdly, the *reasoning coordinator* retrieves the results of global constraint propagation (see point ❸) and includes them in the DL-based descriptions (see point ❹), so that the updated knowledge base can be classified by the DL-based inferential engine.

Fourthly, the *reasoning coordinator* generates queries on local constraints. Based on the feedback from the constraint solver, the *reasoning coordinator* decides whether further reasoning should be carried out and whether there is constraint entailment. Feedback from this stage is included in the DL descriptions which are then used for further DL-based reasoning.

Finally, the *reasoning coordinator* collects the results of the DL reasoner (see point ❺) and sends them to the user via the *user (and KB) interface*. In our example, Organisation1 is deduced to be a subclass of Midsideorg (see the bottom pane of Figure 2), while the numeric constraints are retained as concept hierarchies (i.e. $\mathsf{X}_{ai1} \sqsubseteq \mathsf{X}_{ai}$ and $\mathsf{C}_{org1} \sqsubseteq \mathsf{C}_{midsize}$).

4 Conclusions

We have presented a new approach which fuses the inferences of a taxonomic (DL-based) reasoner with those of a constraint solver without increasing the computational complexity of the former system. This is an exemplary solution to the general problem of reducing reasoning with expressive knowledge to the combination of inferences from heterogeneous reasoners.

Our approach is demonstrated in CONCOR, a hybrid reasoner which combines inferences from both the FaCT DL engine [6] and the Ecl^{i}ps^e constraint solver [3]. The hybrid characteristics of our approach are evident in the polymorphism of *linkages* which are regarded as primary concepts in the DL reasoner and as legal objects (e.g. constrained variables) in the constraint solver. Our approach does not depend on the (DL and constraint) reasoners adopted, since the use of *engine interfaces* and *linkages* [8] makes the *reasoning coordinator* independent from any specific DL system or constraint problem solver.

The CONCOR architecture is part of the $\mathcal{K}-$ShaRe architecture for heterogeneous knowledge sharing and reuse presented in [4]. The CONCOR architecture has been implemented as a standalone reasoning system whose effectiveness has been tested on small examples. Preliminary results are promising and a formal evaluation using larger KBs is forthcoming.

Acknowledgements

This work is partially supported by an Overseas Research Scholarship from the British Council and by the EPSRC under the AKT IRC grant GR/N15764.

References

[1] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proc. of the 12th Intl. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 452–457. Morgan Kaufmann, 1991.

[2] R. J. Brachman et al. Reducing CLASSIC to practice: knowledge representation meets reality. *Artificial Intelligence*, 114:203–237, 1999.

[3] P. Brisset et al. *Ecl^{i}ps^e Constraint Library Manual, Rel. 5.2*. International Computers Ltd. and Imperial College London, 2001.

[4] E. Compatangelo and H. Meisel. $\mathcal{K}-$ShaRe: an architecture for sharing heterogeneous conceptualisations. In *In Intl. Workshop on Intelligent Knowledge Management Techniques (I-KOMAT'2002) – Proc. of the 6th Intl. Conf. on Knowledge-Based Intelligent Information & Engineering Systems (KES'2002) – To appear*.

[5] V. Haarslev and R. Möller. *RACER User's Guide and Reference Manual Version 1.6*. University of Hamburg, Computer Science Department, July 2001.

[6] I. Horrocks. FaCT and iFaCT. In *Proc. of the Intl. Workshop on Description Logics (DL'99)*, pages 133–135, 1999.

[7] B. Hu, E. Compatangelo, and I. Arána. A hybrid approach to extend DL-based reasoning with concrete domains. In *Proc. of the KI-2001 Workshop on Applications of Description Logics*. CEUR Electronic Workshop Proceedings, 2001. Available from http://ceur-ws.org/Vol-44/.

[8] B. Hu, E. Compatangelo, and I. Arana. Inference fusion: a hybrid approach to taxonomic reasoning with numeric constraints. Submitted for publication.

[9] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.