Incorporating Default Characteristics and Approximate Matching in Description Logics

Extended Abstract

Lin Padgham
Dept. of Computer Science,
RMIT University, Melbourne, Australia.

linpa@cs.rmit.edu.au

Description logics are an extremely useful tool in a range of applications where the domain can be described by a hierarchical ontology. The reasoning services of classification, recognition, propagation and consistency checking are valuable in many circumstances. However there are a number of domains and types of application where the strictness of concept descriptions is a limitation. There is no mechanism in description logics for fuzzy matching to a concept description. However this is clearly what we as humans do in many cases.

One of the strengths of description logics is their relationship with logic and their consequent well-defined semantics which provides the basis for correctness and consistency checking. However alternative mechanisms for classification such as clustering algorithms from machine learning (e.g. [Qui93]) are far more able to manage fuzzy matches to descriptions.

There has been a body of work in the last 5-6 years on incorporating defaults into description logics (e.g. [QR92, BH92, BH93, PN93, Str93, Wah96]), motivated partly by the need to describe aspects of concepts that are usually, but not always, true. Although this work includes a number of different approaches, it all focusses on the basic notion that an individual will have default characteristics of its class unless there are reasons for it not to do so. These reasons may be that it is explicitly excepted from the default characteristic, that it belongs to a class that is excepted from the default characteristic, or that it belongs to a class or combination of classes such that the existence of the default characteristic is unclear. None of this work addresses the notion of having a critical mass of the default characteristics of a class, or a clustering around a set of characteristics, even though this seems to be a very basic common-sense notion.[1]

The preliminary work described here explores a way to incorporate a notion of "sufficient default characteristics" into a description logic framework. We have found it important in practice to find ways around the strict logical nature of DL's and we believe that the mechanism introduced here may provide a useful general purpose approach in a wide range of situations. It maintains the advantages of strict logical descriptions, while allowing for a certain degree of fuzziness as to the exact characteristics of members of a class.

The basic idea is to identify a set (or several sets) of default characteristics associated with a particular concept and then to require in the concept description that a member of the concept have at least some number of these default characteristics.

In the following sections we describe the mechanism in some more detail and indicate the way we have used this to advantage in an exploratory application. We note some issues with the model as it currently stands and conclude with a discussion on further work needed.

1 The Representational Mechanism

As in my previous work on defaults in description logics with defaults [], the representation of a concept is split into multiple descriptions. In this model three descriptions are provided in association with each concept - the core concept description which contains the necessary (but not necessarily sufficient) conditions, a default concept description which contains the typical (but not nec-

[1] The application described in [PZ93] does implicitly use this mechanism, but it is not highlighted or discussed in detail.

CSPage-core $\doteq$ Page

CSPage-default $\doteq$
(and (all url-role CS-url)
(fills header (one-of (Computer-Science,
CS, Computer)))
(fills title CS-title)
(all contents CS-contents))

CSPage $\doteq$
(and Page
(at-least 2 cs-default-role))

pageX ::
(and Page
(fills title CS-title)
(fills url-role www.cs.rmit.edu.au)
(fills cs-default-role url-cs title-cs))

Figure 1: Example concept descriptions and individual.

essary) characteristics of the concept, and a basic concept description which provides the additional information regarding to what degree the default characteristics must be fulfilled to allow an individual to be recognised as a member of the concept.

Each characteristic (or top-level clause) in the default description is then given a name which is declared as an individual, and a role *x-default-role* is introduced. The basic concept description then requires at least Y fillers of this role, where Y is the number decided as the critical mass needed.

To clarify this we illustrate with an example. The definitions of the core, default and basic descriptions for a CS-page are shown in figure 1

This example defines a computer science page as being a page with at least two of the default characteristics for a computer science page. Each of the four default characteristics are given a name, which is an individual in the KB - shown in this example are the first and the third clauses which have been given the name *url-cs* and *title-cs* respectively. The individual then has two of the default characteristics and is thus recognisable as a computer science page.

The default description and the core description whilst formally being concept descriptions should really be regarded as auxiliary descriptions, with the basic description representing the named concept.

Given descriptions as in figure 1 it is clearly possible in many if not all cases to automate the process of checking an individual description and asserting the information *(fills cs-default-role title-cs url-cs)*. This could potentially be done as part of the DL engine, and thus tightly integrated with the system, or alternatively it can be done by an external program which makes the determination and asserts the characteristic. We have currently used the latter method for a particular application.

The purpose of the core concept description is to act as a filter for a process of examining an individual description and asserting clauses regarding the default constraints as appropriate. Information that default characteristics of some concept exist in an individual which is not subsumed by the core description of that concept is meaningless and potentially misleading. Consequently our recognition process for individuals uses subsumption by a core concept as a filter for checking the default characteristics of that concept, and asserting appropriate clauses which can then lead to subsumption by the basic concept description.

In our current exploratory use of this mechanism, rather than use the core concept as a filter we have simply checked each individual at assert time for all default characteristics of any concept. This was feasible due to the relatively small number of concepts with default characteristics defined (about 10).

Where an individual's characteristics are modified after the initial assertion, then clearly the assessment as to whether the default constraints were adequately fulfilled would need to be made in connection with each modification. In our exploratory application individual characteristics were never modified and so this was not an issue. However, in principle, rechecking whenever characteristics change can be simply added to other propagation procedures, either withing the DL engine, or as an external mechanism.

In our initial work naming of each default characteristic[2] was done manually and hard-coded into the external program. A more general solution would be to automatically create the individual names on the basis of some part of the characteristic. For example one could use a combination of reserved word, concept name, rolename and "def" to create a suitable string - obtaining names such as *all-r1-X-def, at-least-r1-X-def, fills-r1-X-def, one-of-X-def, same-as-X-def*. Automatic naming of default characteristics would be desirable regardless of whether the checking and assertion was done within the

[2] We have intentionally not defined formally what we mean by a "characteristic", but rather left it to intuition. This needs to be examined in more detail in connection with further automation of the process.

DL kernel or externally. It would of course be necessary if it was done within the kernel, unless one added some syntactic facility to the language to allow manual naming.

The current limited use of this mechanism for a particular application has been implemented using standard subsumption and recognition procedures, but with a mechanism for examining individuals and automatically asserting information regarding their default characteristics. There are a number of possibilities regarding both implementation and formal definition of the mechanism for the general case. There are issues regarding subsumption and possible missing of subsumption relationships between arbitrary concepts and the basic concept description which contains the constraint regarding the degree to which the default needs to be fulfilled. Both implementation and semantics of possible definitions of the basic idea require substantial further investigation. This further investigation does appear to be warranted by the usefulness of the mechanism in our application.

2 Initial Application

The application which prompted concretising of some of these ideas was an application to describe and categorise pages on the WWW. Our aim was to develop descriptions of types of pages within a particular domain (we used a computer science web site), categorise pages within a site according to these descriptions and then use this categorisation for conceptual querying of the domain. Examples of types of pages were things like *Subject-description-page*, *Research-project-page* and *Academic-staff-page*.

As is common in many domains there were relatively few characteristics that could categorically be attributed to types of pages (beyond membership in particular superclass(es)). However there were characteristics which were typical for particular types of pages (at least within a given site) and if sufficient number of such characteristics were present then this was reasonable evidence for the page to belong to the category in question.

A simple version of the mechanism outlined was used to describe the relevant concepts and individuals. This produced considerable success in categorising page types correctly. Without some such mechanism it would not have been possible to classify pages adequately.

3 Discussion and Further Work

Most work on defaults, both within Description Logics and otherwise focusses on attribution of default characteristics to an individual as long as there is no reason not to do so, rather than on use of default characteristics to recognise individuals. Recognition of individuals by virtue of default characteristics is addressed to some extent in [PN93] and is the main focus in the work of Coupey and Fauquere [CF95]. However both these approaches require dealing with exceptions and do not support a model requiring some critical mass of default characteristics for recognition of a concept instance.

This approach, whilst focussing on recognition via some threshold number of characteristics does not address the issue of assertion of default characteristics where reasonable. This is an area for further work and while automatic attribution of default characteristics will undoubtedly have the same issues (both semantic and computational) as in other models, it seems likely that some limited use could be made of the information existing in the default concept descriptor regarding assumptions about unstated characteristics of individuals.

As noted previously there are a number of possibilities regarding formal definition of the basic ideas. These need to be guided both by implementation considerations and by semantics. For example one could clearly achieve a similar effect as described above by simply using a language incorporating "**or**" and describing concepts in terms of different possible combinations of default characteristics. However the price paid for this in determining subsumption makes it unusable in practice for many applications.

Acknowledgements

The author thanks AT&T Bell Labs for the use of the CLASSIC system and Patrick Lambrix for useful discussion regarding these ideas. The author also thanks Van Anh Bui and Nathan Lambert for their efforts in implementing the WebKB project where some of the ideas described here were trialled, and Elisabeth Sonenberg for her support of and contributions to the WebKB project.

References

[BH92] Franz Baader and Bernhard Hollunder. Embedding defaults into terminological knowledge representation formalisms. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference*, pages 294–305, Cambridge, MA, 25–29 October 1992.

[BH93] Franz Baader and Bernhard Hollunder. How to prefer more specific defaults in terminological default logic. In *Proceedings of the Inter-*

national Joint Conference on Artificial Intelligence - IJCAI 93, pages 669–674, 25–29 October 1993.

[CF95] Pascal Coupey and Cristophe Fouquere. Defaults in concept definitions: An application. In *International KRUSE Symposium: Knowledge Retrieval, Use and Storage for Efficiency*, pages 112–121, Santa Cruz, CA, USA, AugAugust 1995.

[PN93] Lin Padgham and Bernhard Nebel. Combining classification and non-monotonic inheritance reasoning: A first step. In *Proceedings of Seventh International Symposium on Methodologies for Intelligent Systems*, pages 132–141, Trondheim, Norway, June 1993. LNAI 689.

[PZ93] Lin Padgham and Tingting Zhang. A terminological logic with defaults: A definition and an application. In *13th International Joint Conference on Artificial Intelligence*, volume 2, pages 662–668, Chambéry, France, 29–3 September 1993.

[QR92] Joachim Quantz and Véronique Royer. A preference semantics for defaults in termonological logics. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd International Conference*, pages 294–305, Cambridge, MA, 25–29 October 1992.

[Qui93] J.R. Quinlan. *C4.5, Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[Str93] U. Straccia. Default inheritance reasoning in hybrid kl-one-style logics. In *Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI 93*, pages 676–681, 1993.

[Wah96] Niclas Wahlöf. A default extension to description logics and its applications. Lic. thesis 591, Department of Computer and Information Science, University of Linköping, Sweden, 1996.