Capability Management and Discovery in Description Logic

Dong Cheng and Nacer Boudjlida

LORIA-University Henri Poincaré Nancy 1
BP 239, 54506 Vandoeuvre Lès Nancy CEDEX(F)
Email: cheng,nacer@loria.fr

Abstract. In some application domains as composite software, ranging from search engines, to more general applications, like cooperative and distributed applications or e-business and e-commerce applications, more and more rely on capability description. In this paper we try to highlight some concepts concerning capability description and discovery, and we discuss the relationship between capabilities. Specifically, we choose a capability description language, as and extension of $\mathcal{ALN}$ Description Logics languages. The mediation services are expected to play an important role in helping automated processes to access heterogeneous information. They are not Yes/No answers from the mediator, when no single object meets the search criteria, they may be cooperative answers to make a composite answer.

1. Introduction

The propagation of the Web Services application has led to increasing need for network services discovery. As each business registered with UDDI categorizes all of its Web services according to a defined list of service types. Businesses can search the registry's listed service types to find service providers. The *tModel* is an abstract data structure representing the capability of business service providers. On the other hand, in some situations we need a composite service by multiple service providers to satisfy a complex requirement. The capabilities of service providers need be described in formal knowledge description languages. Allen Newell [1] has analyzed what he terms the *knowledge level* and he has situated knowledge in the epistemological processes of an observer attempting to model the behavior of another agent. That is knowledge is ascribed to an agent to explain its capabilities, and there is no knowledge without capabilities.

In our work, we define a method for capability management, and then we apply the method for capability discovery and composition in a distributed heterogeneous knowledge base. The heterogeneous knowledge base might use different formal description languages, as Description Logics, Frame-logic, etc.

The presentation is structured as follows. In section 2 we briefly review the conceptual federated mediator-based architecture, and the capability description language. In section 3, we describe some concept on capability discovery. Conclusions and remarks are in section 4.

2. Capability Description in Mediation Architecture

In the mediator-based architecture [3], one should notice that some cases conduct to a failure of the query when only one mediator is involved. But, if we assume a grouping of mediators (into a federation of mediators); these cases are typical cases where cooperation among the mediators is required. When a mediator partner dissatisfies the query, we need to determine "what is missing?" to the "entities" to satisfy query. That means to determine what part of the query is not satisfied by the found "entities". That part as well as the original query is transmitted then to a mediator of the federation. Conceptually, we can see the query as being addressed to "the union" of by the federated mediators' knowledge bases. The query evaluation and the composition of an answer are performed thanks to the federated mediators, every mediator having its proper Capability KB. This semantic query describes the services or the capabilities an "entity" might offer. The capability of an "entity" is presented under formal concepts. In many situations, it is not possible to find any "entity" which exactly provides the expected. So we need a kind of method to exactly describe the capability itself in the context of semantic queries.

In our previous work, we defined a mediator-based architecture where heterogeneous systems may work together in the context of semantic queries [2]. We adopted a *Description Logics* (DLs) language to represent the semantic query. DLs are a family of knowledge representation languages that is intensively developed and studied in the field of Knowledge Representation. In DLs a description of a world is built using *concepts*, *roles* and *individuals*. The *concepts* model classes (sets of *concepts*, called Terminologic Box or TBox) of individuals (sets of *individuals* are called Assertion Box or ABox). *Capability* is described on *roles* (unitary attribute and binary relationship) in this kind of description structure.

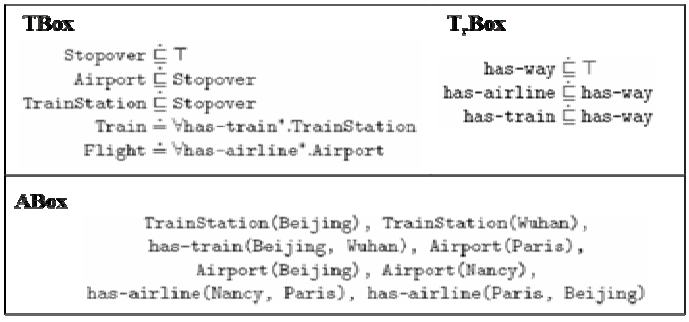

Figure 1. An example of Knowledge Base in $\mathcal{ALN}_{\text{role+}}$

Some extended role descriptions were mentioned and proved in [4]. The added value of transitive closure in individual capability composition is shown in [5]. For example, an $\mathcal{ALN}_{\text{role+}}$-concept description is $\texttt{Flight} \doteq \forall \texttt{has-airline}^*.\texttt{Airport}$ which intuitively describes all air travel that includes Non-stop flights and Transfers. In particular, $\forall \texttt{has-airline}^*$ is interpreted as reflexive-transitive closure of the role **has-airline**, thus representing the role "transfers" as showed in figure 1.

Intuitively, this ABox in figure 1 says that {**Paris**, **Beijing**, **Nancy**} have {**Airport**}, and there are {**Airline**} from {**Nancy**} to {**Paris**}, and from {**Paris**} to {**Beijing**}. So we can find that there exist flights from {**Nancy**} to {**Beijing**} with very simple reasoning by composing individuals' capabilities. We will introduce T_rBox in next section.

3. Capability Discovery

As mentioned in section 2, we use *role* to present entities' capability where the *role* is binary relationship between two entities. But many current research work focus on the concept description and proposition reasoning. As we known that TBox and ABox are the two main components of knowledge base in terminology language. The *role* just is an assisted part of *concept* description. In this work, we introduce a new concept, capability space (T_rBox), into the knowledge base. *Role* description is used in a T_rBox to define the roles of the application domain. We can imagine a knowledge base as drawn in figure 1. A capability space, T_rBox, is introduced in this knowledge base: it contains the descriptions of the roles and those of the relationships between roles. In the travel example, the following T_r-Box is added to the knowledge:

Three relationships, **has-way**, **has-airline** and **has-train**, exist in this T_rBox. As we see, the concepts are often organized into a concept hierarchy by the subsumption relationship in TBox. Then we may implement some knowledge reasoning services on the concept hierarchy by the *classification* approaches, as subsumption relationship satisfaction [3], complement concept determination [2], etc. We may also organize the roles into a hierarchy of roles by the subsumption relationship.

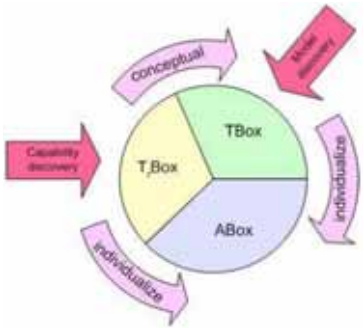

Figure 2. Request action in Knowledge Base using T_rBox

There are three connections in this knowledge space: concept-individual, concept-role and role-individual, as drawn in the figure 2. We defined a request action for capability discovery in a federation of mediators based on the subsumption relationship. As example in figure 1, we want to apply a capability discovery on **has_way(Nancy, Wuhan)**. We can find the capability **has_way** in T_rBox, and who has two sub-capability **has_airline** and **has_train**. Then we can conceptualize this result in TBox, there are two concepts **Flight** and **Train** who implement this capability. In the end, we can individualize this result in ABox. We can find a way from **Nancy** to **Wuhan**. There is a composite answer. Firstly, a flight from **Nancy** to **Beijing**, and who transfer in **Paris**. Secondly, it exist a train from **Beijing** to **Wuhan**.

4. Conclusion Remarks

We believe that capability discovery and composition will be more and more applied in knowledge discovery and management domains. We believe that the hard problem do not go away even if we solve low-level issues such as defined relationship analysis, complex term mapping, and ontologies integration

Based on our initial experience, there is formal relationship between capabilities and common mathematical logic background knowledge, so two interrelated approaches have been paid attention in our work. The first one is finding some general rules between defined capability relationships which are described in a formulaic language. The second one is defining the capability discovery approaches and finding possible capability compositions. In the future, we may consider to design and implement a platform, where systems will accept a capability discovery query, and it can support heterogeneous knowledge representation technologies.

References

1. Newell, A.: The knowledge level. Artif. Intell. 18(1) (1982) 87–127
2. Cheng, D., Nacer, B.: Federated mediators for query composite answers. In: 6th International Conference on Enterprise Information Systems - ICEIS'2004, Porto, Portugal. Volume 4 (2004) 170–175
3. Boudjlida, N.: A Mediator-Based Architecture for Capability Management. In Hamza, M., ed.: Proceedings of the 6th International Conference on Software Engineering and Applications, SEA 2002, MIT, Cambridge, MA (2002) 45–50
4. Baader, F.: Description logic terminology. In Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003) 485– 495
5. Baader, F., Küsters, R.: Unification in a description logic with transitive closure of roles. In: LPAR '01: Proceedings of the Artificial Intelligence on Logic for Programming, London, UK, Springer-Verlag (2001) 217–232