STEREO: a SaT-based tool for an optimal solution of the sERvice selEctiOn problem

Daniel Izquierdo, María-Esther Vidal, and Blai Bonet

Departamento de Computación
Universidad Simón Bolívar
Caracas 89000, Venezuela
{idaniel,mvidal,bonet}@ldc.usb.ve

Abstract. We present STEREO, a system that offers an expressive formalism and implements techniques firmly grounded on logic to solve the Service Selection Problem (SSP). STEREO adopts the Local-As-View approach (LAV) to represent services' functionality as views on ontology concepts, while user requests are expressed as conjunctive queries on these concepts. Additionally, users can describe their preferences, which are used to rank the solutions. We discuss the LAV formulation of SSP; then, we illustrate the encoding of SSP as a logical theory whose models are in correspondence with the problem solutions, and in presence of preferences, the best models are in correspondence with the best-ranked solutions. We demonstrate STEREO and the properties of modern SAT solvers that provide an efficient and scalable solution to SSP.

1 Introduction

Nowadays, several annotation tools like the one proposed by Ambite et al. [1], are able to label and convert existing Web data sources into Semantic Web Services. Once this tremendous amount of services becomes available, users will more than ever require approaches to precisely express their selection needs and to effectively identify the services that best meet their requirements. To achieve this goal, we developed STEREO, a system that offers an expressive formalism for describing Web services and user requests, and implements a logic-based solution to efficiently select the services that best meet the user requirements. The STEREO formalism and techniques have been reported in [4].

STEREO is tailored to constantly changing Web service datasets and a relatively stable set of ontology concepts. STEREO adopts the recent approach of Ambite et al. [1] that describes services as views on ontology concepts following the LAV approach that is widely used in integration systems [5]. Thus, every time a service changes or a new one becomes available, only a tiny fraction of the mappings must be updated. Additionally, STEREO models user requests as conjunctive queries on the ontology concepts and casts SSP as the problem of rewriting a query in terms of a set of views, the so-called Query Rewriting Problem (QRP). Thus, STEREO exploits existing scalable and efficient techniques that have been proposed in the data integration area [2, 5]. Furthermore,

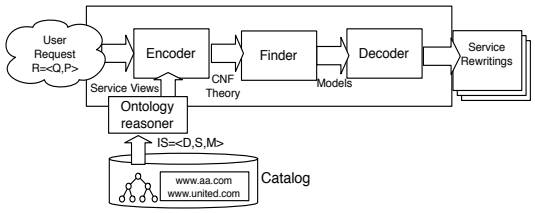

Fig. 1. The STEREO Architecture

STEREO offers a simple yet expressive language for preferences and supports users' preferences and constraints on the set of the selected services to satisfy a given request. These preferences and constraints further refine and rank the set of valid rewritings of the posed query, in a way that the best solutions to SSP corresponds to the best-ranked valid rewritings of the corresponding QRP. STEREO constructs a propositional logical theory that captures the features associated with SSPs; each model of the theory encodes a valid combination of services, and these models are ordered by their rank and the best-ranked models are the models with minimum rank. Knowledge encoded in the ontology is used to extend the rules in the logical theory to permit the cover of ontology symbols in the query with symbols in the views according to subsumption relationships represented in the ontology. We demonstrate the benefits of the approach and show the following key issues: the expressiveness by modeling a real-world domain; the scalability by demonstrating how STEREO is able to enumerate realistic instances of SSP in a few seconds; and finally, the performance by enumerating in the same logical theory, the optimal models based on different cost/rewards of the users' preferences. The demo is published at `http://stereo.ldc.usb.ve/STEREO`.

2 STEREO Architecture

The STEREO architecture is comprised of a Catalog of service descriptions, an Ontology Reasoner, the Encoder, the best model Finder, and the Decoder. Figure 1 depicts the overall architecture. An instance of SSP consists of an integration framework IS and a user request R. Formally, IS is a tuple $\langle D, S, M\rangle$ where D is an ontology[1], S is the set of services, and M is the set of LAV mappings. R is a tuple $\langle Q, P\rangle$ where Q is a conjunctive query on the ontology relational symbols and a set of preferences P. The STEREO Catalog is populated with the integration system components. We developed an Ontology Reasoner to compute the transitive closure of the subsumption relationships, and the Encoder module makes use of this information in conjunction with the LAV mappings, constant symbols, and users' preferences to encode an input instance of SSP

[1] An ontology is comprised of a set of relational and constant symbols from which logical formulas can be constructed, and a collection of axioms describing the ontology.

as a CNF theory. STEREO uses c2d (`http://reasoning.cs.ucla.edu/c2d`) to compute all the best-ranked models of the CNF theory in a certain normal form called deterministic and decomposable negation normal form (d-DNNF) [3]; however, off-the-shelf SAT solvers can be also used. The Finder computes a best model by enumerating the models and it can reuse the same d-DNNF for different cost/rewards associated with the preferences to calculate the best service rewritings. The Decoder translates the model(s) into the input instance.

3 Demonstration of Use Cases

We illustrate our approach on a simple travel domain that contains information about flight and train trips between cities and information about which cities are in the US. The ontology is comprised of the predicates *trip*, *flight*, *train* and *uscity*. The first predicate relates cities (x, y) if there is a direct trip either by plane or train between them. The *flight* predicate relates (x, y, t) whenever there is a direct flight from x to y operated by airline t, and similarly for *train*, and *uscity* indicates if a given city is or not a US city. The ontology axioms capture two subsumption relations:

$$\mathit{flight}(x, y, t) \sqsubseteq \mathit{trip}(x, y).\ \mathit{train}(x, y, t) \sqsubseteq \mathit{trip}(x, y).$$

For the services, we assume the following type of data sources:

- *nat-flight*(x, y, t) relates two US cities that are connected by a direct flight,
- *one-way-flight*(x, y) relates two cities that are connected by a one-way flight,
- *nat-train*(x, y) relates US cities that are connected by a direct train,
- *to-pa*(x) tells if there is a direct flight from x to Paris operated by American,
- *from-la*(x) tells if there is a direct flight from LA to x operated by United.

Based on the ontology concepts the services are described using LAV views:

$$\begin{aligned}
\mathit{nat\text{-}flight}(x, y) &:- \mathit{flight}(x, y, t),\ \mathit{uscity}(x),\ \mathit{uscity}(y)\,. \\
\mathit{one\text{-}way\text{-}flight}(x, y) &:- \mathit{flight}(x, y, t)\,. \\
\mathit{nat\text{-}train}(x, y) &:- \mathit{train}(x, y, t),\ \mathit{uscity}(x),\ \mathit{uscity}(y)\,. \\
\mathit{to\text{-}pa}(x) &:- \mathit{flight}(x, \mathrm{Paris}, \mathrm{AA})\,. \\
\mathit{from\text{-}la}(x) &:- \mathit{flight}(\mathrm{LA}, x, \mathrm{UA})\,.
\end{aligned}$$

Consider a user that needs to select the services able to retrieve one-stop round trips from a US city x to any city y in the world, and the user prefers flying than traveling by train. This request is represented as follows:

$$Q(x, y) :- \mathit{uscity}(x),\ \mathit{trip}(x, u),\ \mathit{trip}(u, y),\ \mathit{trip}(y, v),\ \mathit{trip}(v, x)\,.$$

This preference is modeled by assigning a high reward to the symbol *flight*. Any rewriting of the ontology predicates in terms of the services defined by these predicates, implements the request. In addition, rewritings comprised of services

defined by the symbol *flight* will have higher scores than those with services defined by the symbol *train*. Thus, this rewriting is valid and highly ranked:

$$I(x, \text{Paris}) \ :- \ \textit{nat-flight}(x, u),\ \textit{to-pa}(u),\ \textit{one-way-flight}(\text{Paris}, v),\ \textit{nat-flight}(v, x)\,.$$

But, the following rewriting is not a valid solution because it maps the query variable y into two different constants Paris and LA that denote different cities:

$$I'(x, y) \ :- \ \textit{nat-flight}(x, u),\ \textit{to-pa}(u),\ \textit{from-la}(v),\ \textit{nat-flight}(v, x)\,.$$

Using this travel domain, we consider the following scenarios:

- We run STEREO in three different benchmarks. We start with a benchmark of queries with 2 to 5 sub-goals and sets of 10 to 100 services; we show that the sets of 100 airlines with 5-stop flights can be compiled in 328 secs, the best model can be computed in 0.29 secs, and the enumeration of all models in 0.47 secs. We add a second service for each airline and show that STEREO behavior remains similar. Finally, we try services with multiple sub-goals which are randomly generated and show that the compilation time does not grow monotonically with the number of views. The time to find the best model is 0.46 secs while the enumeration of all models is about 17 hours.
- We test the system with ontologies of different sizes and show reusability by assigning different cost/reward to the preferences; we also demonstrate that the same d-DNNF can be used to compute the best model, while the execution time remains almost the same.

4 Conclusions

We present a logical-based approach that relies on the LAV formalization to solve SSP; the approach is expressive, efficient and scalable. We illustrate the formalization of service functionalities in terms of simple conjunctive rules. In addition, we demonstrate a propositional logic-based formalization of SSP instances, and different scenarios where the properties STEREO can be observed by users. We show how the approach can be applied to real-sized problems, while the whole approach is only possible when the compilation of the CNF theory into d-DNNF succeeds.

References

1. J. L. Ambite, S. Darbha, A. Goel, C. A. Knoblock, K. Lerman, R. Parundekar, and T. A. Russ. Automatically constructing semantic web services from online sources. In *ISWC*, pages 17–32, 2009.
2. Y. Arvelo, B. Bonet, and M.-E. Vidal. Compilation of query-rewriting problems into tractable fragments of propositional logic. In *AAAI*, 2006.
3. A. Darwiche. Decomposable negation normal form. *J. ACM*, 48(4):608–647, 2001.
4. D. Izquierdo, M.-E. Vidal, and B. Bonet. An expressive and efficient solution to the service selection problem. In *ISWC-To Appear*, 2010.
5. R. Pottinger and A. Y. Halevy. Minicon: A scalable algorithm for answering queries using views. *VLDB J.*, 10(2-3):182–198, 2001.