Query Rewriting for Semantic Web Information Integration

Dave Kolas

BBN Technologies
1300 N 17th St., Suite 400, Arlington, VA, U.S.A.
dkolas@bbn.com

Abstract

Though significant research has been done in the realm of information integration, continuously increasing amounts and complexity of data demand more advanced techniques for its access. Semantic Web technologies such as OWL and SWRL provide the capacity for rich definitions of data sources, global views, and the mappings between them. An information integration system based on these technologies is thus very attractive; however, OWL and SWRL would appear to introduce significant complexity to query reformulation. We motivate our work via a distributed query system architecture based on Semantic Web technologies, and discuss the application of Global as View and Local as View approaches to this scenario. Finally, we introduce modifications to the Global as View approach for this system and analyze their effects.

Introduction

As the amount of data available on the Web and from other sources explodes, there is an ever increasing need for more advanced data integration techniques. Organizations that assimilate this information for processing are left drowning in the sheer volume of this data. An organization's perspective on available data is constantly evolving, and as such their information integration systems must also evolve. New sources of data appear constantly, and existing sources change or disappear. If the information integration systems cannot quickly adapt to these types of changes, their value becomes negligible.

Significant advances in the realm of information retrieval have made searching the text of documents significantly more effective. However, these techniques are primarily focused on unstructured documents and documents whose structure are designed for human visualization, and cannot hope to do more than point the consumer to relevant documents. As such, they are not well suited for answering structured queries over structured sources.

The goal of the Semantic Web is to bridge this gap by making information on the Web interpretable by computers. Common languages such as RDF [RDF], OWL [OWL], and SWRL [SWRL] were created to allow rich data definition in a standard language that could be processed in a well-defined way. OWL, based in Description Logic, and SWRL, based in Description Logic and Logic Programming, when used together provide a basis for far more expressive data descriptions than are currently possible with web data formats such as XML.

This richness of data definition offers significant benefits for information integration. Queries can be expressed in a more abstract fashion, and more domain knowledge can be encapsulated in the data definition. This means that the information consumer spends less time determining the appropriate vocabulary for their query and thus has more time to take action on the results.

The goal of this paper is to apply existing work on information integration systems to an information integration system using Semantic Web technologies. This system will use OWL ontologies instead of schema definitions and SWRL rules instead of view definitions. We will examine exactly what is desired from such a system, and then explore how current approaches can be extended to achieve those goals.

The rest of the paper will be structured as follows. First, we will consider the larger picture of an architecture in which this type of semantic query rewriting would be necessary. Next, we will attempt to apply standard techniques for schema mediation, Global as View and Local as View, to this scenario, analyzing the strengths and weaknesses of each approach. Finally, we will propose modifications of the Global as View approach which overcomes the weaknesses, and analyze their effects on query rewriting.

Semantic Distributed Query Architecture

The semantic distributed query architecture is motivated by a desire to provide data integration through flexible mappings, created in the form of ontologies and rules. The system makes use of both Description Logic-based descriptions in OWL and Horn-like rules in SWRL. This leads to two specific aims for the system design. First, make use of the expressivity of OWL and SWRL to allow for a level of conceptual data independence beyond that which can be provided by non-semantic descriptions. Second, the mapping between the data sources and the semantic system should be done in such a way that the underlying data storage mechanisms are completely irrelevant to the consumer.

To facilitate these goals, the architecture splits the query and data transformations into two distinct tiers of processing (See Figure 1). The upper tier of the architecture is responsible for ontology to ontology transformations, subquery ordering, and information integration, commonly referred to as the *mediator*. The component that fulfills this role is referred to as Semantic

Query Decomposition (SQD). The lower tier is responsible for encapsulating native data sources as ontological knowledge with a SPARQL [SPARQL] interface. This tier must map SPARQL queries into queries appropriate for a particular native data source (e.g. SQL for relational databases). This tier is manifested in Semantic Bridges for each type of data source, and corresponds with the well known concept of a *wrapper* [Ullman97]. We will now look at SQD and two instances of Semantic Bridges in more detail.

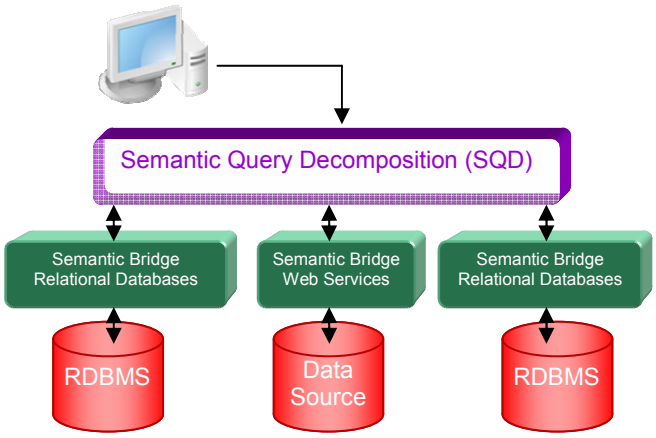

Figure 1

Semantic Query Decomposition

The goal of the upper tier of the architecture is that of the mediator in standard information integration systems. This involves receiving the query, deriving a query plan over the various sources, and optimizing and executing this plan. Unlike a relational system, however, the set of inputs to this process are based in Semantic Web technologies. A query Q is defined in the language SPARQL, using terms from the *domain ontology, O,* defined in OWL. Data source descriptions are defined as *data source ontologies*, $D_1...D_n$, also defined in OWL. Mappings from the data source ontologies into the domain ontology $M_1..M_n$ are expressed as sets of rules defined in SWRL. This mediator then must derive a set of subqueries $Q_{x1}...Q_{xm}$ where $x_1..x_m \in \{D_1...D_n\}$ in SPARQL whose combined results are equivalent to those of Q.

Semantic Bridge for Relational Databases

One instance of a Semantic Bridge is the Semantic Bridge for Relational Databases (SBRD). This component is designed to connect to a relational database and present its contents according to a data source ontology. This implies defining a mapping between the data source ontology and the underlying relational structure. Work has been done by the Semantic Web community in mapping SPARQL queries over a virtual RDF graph to SQL queries over a relational database given an RDF to RDBMS mapping [D2RQ], and the process for doing such is beyond the scope of this paper.

Semantic Bridge for Web Services

Another instance of the Semantic Bridge is the Semantic Bridge for Web Services (SBWS). The purpose of this component is to serve as a wrapper over a web service with an interface defined by WSDL. Unlike SBRD, SBWS is not mapping one general query language to another, but rather mapping one general query language onto a set of specific predefined operations. This has proven challenging, though precedent does exist [LRO96]. Again, we will assume that such a thing is possible and that the only relevant problem for the mediator is producing an appropriate SPARQL query.

Applying Mediated Schema Approaches

Two primary approaches [Levy00] exist for restricting the mappings between local schemas and the target schema in order to make the query rewriting process more tractable. These are Global-as-View (GAV), where elements in the target schema is defined as a view over the local schemas, and Local-as-View (LAV), where elements in the local schemas are defined as a view over the global schema. We will examine each of these in turn, analyzing how each applies to our semantic distributed query architecture, and the practical downsides to each. First, however, it is useful to examine the common elements of applying these approaches to Semantic Web information integration system.

Because SWRL as an addition to OWL is essentially datalog restricted to unary and binary predicates, much of the literature on information integration approaches applies to it quite directly. Some of the vocabulary differs, though: whereas datalog descriptions generally refer to *predicates*, SWRL rules, with their heritage in Horn rules, generally refer to *atoms*. Though SWRL is technically a superset of OWL, most of the translation rules that have been created in practice make use of this datalog portion of the language. As such, that will be the focus of discussion in this paper, excepting the section on future work.

Thus, these SWRL rules directly correspond to the view definitions found in the literature on both the GAV and LAV approaches. One exception to this correspondence is that SWRL rules are permitted to have multiple atoms in the head (consequent):

$$Pie(x) \wedge containsFruit(x, y) \Rightarrow Dessert(x) \wedge containsIngredient(x,y)$$

However, this rule structure does not add any complexity to the processing, as these rules can simply be rewritten as multiple rules with single-atom consequents:

$$Pie(x) \wedge containsFruit(x, y) \Rightarrow Dessert(x)$$

$$Pie(x) \wedge containsFruit(x,y) \Rightarrow containsIngredient(x,y)$$

Now we will examine how the GAV and LAV approaches apply to a Semantic Web system individually.

Global as View

Applying the GAV approach to our system based on OWL ontologies and SWRL rules means treating the SWRL rules as view definitions as above. In the GAV approach, each mediated schema predicate is implied by a conjunction of data source predicates. This means that for every predicate in the domain ontology, a direct derivation is required directly from data source concepts.

The GAV approach has been shown to have both positive and negative aspects with respect to information integration systems [Levy00], and we believe these apply equally to a Semantic Web based information integration system. On the positive side, query rewriting is straightforward, and since the portion of SWRL used for the mappings is a subset of datalog, this is true in this scenario as well. Essentially all that needs to be done is replacing mediated schema predicates in the query with the data source predicates that imply them. Also, GAV statements follow a natural mapping paradigm; they follow the format of *if* some set of predicates are true in the data source, *then* some set of predicates are true in the mediated schema.

On the other hand, GAV has some significant disadvantages. First, the mediated schema is dependent upon the data source schemas for structure. Also, mappings from the data sources to the mediated schema are not mutually independent. This is generally thought to limit the scalability of the GAV approach, and it is equally limiting when using Semantic Web technologies.

Local as View

In the LAV approach, the mapping descriptions are reversed. The contents of the data sources are described in terms of predicates on the mediated schema. Once again, we could use SWRL rules to represent these relationships. A LAV approach has been proven successful in a Semantic Web context, though it made use of only OWL predicates for alignment and not SWRL rules [DHQW06].

The LAV approach has two major advantages over the GAV approach. First, the mediated schema is more independent. Because local sources are defined in terms of the mediated schema, it is possible to start with whatever mediated schema is desired. Perhaps more importantly, the data source mappings are mutually independent. There is no interaction between data sources in the definitions of the source mappings, and thus a new source can be added, modified, or removed without affecting any of the other mappings.

Unfortunately, this approach makes query rewriting significantly more complex. Algorithms for this rewriting have been developed [Levy00, PH01], but even then recursive queries may be required [Levy00].

Information Integration Goals

There are several criteria that define a successful approach to the data integration problem in this context. Each facilitates practical usability in a real-world scenario. We will consider several of these and discuss why each is desirable.

First, the mediated schema should be defined independently. There are many reasons why this is desirable. If the mediated schema is built independently, it is much less likely to reflect the biases of any of the individual sources relative to one another. It is also more likely to be free of the limitations of whatever underlying access mechanism is being used for each of the sources. Moreover, the users' perspective is likely to change over time. Keeping the mediated schema independent allows change of this perspective while minimally affecting the data source mappings.

Second, mappings from each source to the mediated schema should be mutually independent. This is critical, as any information integration system is only going to grow in complexity over time. As more useful sources of data are discovered, the complexity of adding them cannot increase or the approach will not be scalable. Over time, it is also likely that sources will be replaced with updated versions of themselves. If removing, adding, or replacing a data source requires revisiting the mappings between every data source and the target, maintenance of the system would become unwieldy very quickly.

Third, the mappings should be able to leverage all of the meaning stored in the sources. There tends to be a mismatch between the Semantic Web technologies, which are primarily based on monotonic reasoning and an open world assumption, and typical relational database implementations, in which the absence of values can often be interpreted as having meaning. While this is not necessarily a general information integration problem, it is a problem that the system will have to overcome in order to be effective.

Finally, the mappings should be as straightforward to create as possible. Ideally, these mappings could be created by a domain expert and not require a software engineer. While this is notably more difficult to quantify than the previous goals, and undoubtedly a matter of opinion, it is our belief that writing source to target transformations is easier than defining sources in terms of the global schema.

Modified GAV Approach

With the desire for an independent global schema and independent source mappings, it may seem as though we are trending towards an approach based on LAV. However, we feel that the desire to write the mappings as source-to-target is extremely important, and perhaps more importantly that the other considerations can be accommodated within a modified GAV approach.

It is worth noting that it has been shown that under certain circumstances one need not choose between LAV and GAV approach. In the GLAV approach, mappings can be defined in either direction, provided the global schema meets certain conditions [FLM99, CCGL02].

Unfortunately, the primary condition that the global schema has to meet for GAV and LAV to be interchangeable is the inclusion of integrity constraints. Since OWL and SWRL have no capacity for integrity constraints, this requirement could not be met without fundamentally altering the spirit of OWL/SWRL based reasoning. Other approaches exist for combining them as well, but are based on relational algebra and thus less directly applicable to this scenario [XE].

With these things considered, we now present an application of modified GAV to a Semantic Web information integration system. For the purposes of illustration, we will use a simple example, with three data sources, and a domain ontology:

Pie Supplier: *(pie:)* "Pete's Pies" is a commercial suppler of pies

Dessert Shop: (shop:) "Dave's Desserts" is a retail outlet selling pies and other items, and purchases their pies from Pete's.

Delivery Service: (del:) "Don's Delivery" is a delivery service that works with Dave's Desserts as well as other retail stores

Domain Ontology: (pa:) represents the mediated schema for the three sources, assumed to be developed by an organization of pie aficionados

A full description of the predicates in these three ontologies is in the Appendix. Note that this scenario assumes an ideal world in which pie can be delivered to one's door.

Independent Data Sources

Perhaps the most glaring problem with applying GAV to an information integration system is non-independent source to target mappings. When only one data source is required for a given predicate in the domain ontology, this is not a problem. Consider the mapping:

[Mapping1] $pie{:}Pie(x) \wedge pie{:}containsFruit(x,\text{"apple"}) \Rightarrow pa{:}ApplePie(x)$

This mapping is likely to be unaffected by the addition, deletion, or modification of other data sources. However, rules that come from multiple sources such as these may not be so easy:

[Mapping2] $pie{:}Pie(p) \wedge pie{:}containsFruit(p,f) \wedge pie{:}name(n) \wedge shop{:}Dessert(d) \wedge shop{:}name(d,n) \wedge del{:}Order(o) \wedge del{:}productOrdered(o,p) \wedge del{:}orderedFrom(o,s) \wedge del{:}Shop(s) \wedge del{:}shopName(s,\text{"Dave's Desserts"}) \wedge del{:}Product(p)\ del{:}productName(p,n) \Rightarrow pa{:}stocksPieWithFruit(s,f)$

[Mapping3] $pie{:}Pie(p) \wedge pie{:}containsFruit(p,f) \wedge pie{:}name(n) \wedge shop{:}Dessert(d) \wedge shop{:}name(d,n) \wedge del{:}Order(o) \wedge del{:}orderedBy(o,c) \wedge del{:}Customer(c) \wedge del{:}productOrdered(o,p) \wedge del{:}orderedFrom(o,s) \wedge del{:}Shop(s) \wedge del{:}shopName(s,\text{"Dave's Desserts"}) \wedge del{:}Product(p) \wedge del{:}productName(p,n) \Rightarrow pa{:}likes(c,f)$

Mapping2 states that if the delivery service has delivered a pie from Dave's Desserts that contains a fruit, the store stocks a pie with that fruit. Mapping3 states that if a customer has ordered a pie that contains a fruit, the customer must like that fruit (the pie aficionados are unyielding in their belief that no person orders a pie with a fruit that they do not at least subconsciously like). This type of rule is not necessarily a problem with only three sources, but when more similar sources are added, the complexity explodes. Adding another pie supplier with the same schema means doubling the number of mappings like Mapping3. Adding another pie supplier, another dessert shop, and another delivery service means Mapping3 has now become eight different mappings. The maintenance cost increases further if the schemas for these new sources are subtly (or vastly) different.

These complications are derived from the restriction noted before that all antecedent terms in the GAV approach must be derived only from data source predicates. By lifting this restriction and imposing two new restrictions, we can allow the data source mappings to be defined independently.

[R1] All predicates appearing in the antecedent of mapping $m \in M_d$ are in $O \cup D_d$

[R2] All predicates appearing in the consequent of mapping $m \in M_d$ are in O

This implies that while each mapping's consequent must still be expressed in the predicates of the domain ontology, its antecedent may now be comprised of predicates both from the associated single data source and the domain ontology. However, predicates from multiple data sources are no longer allowed to appear in the same mapping.

Queries could only be answered, however, if these mappings are acyclic:

[R3] For each predicate used in the mappings, assign a node in a graph. For each pair of predicates such that one appears in the antecedent and one appears in the consequent of the same mapping, assign a directed link from the antecedent node to the consequent node. The resulting graph must be acyclic.

Since this condition can be checked for at design time, queries stuck in recursive loops can be prevented. It is easy to see then that mappings of this type can be "unfolded" into view definitions in the standard GAV approach.

The implications to the maintenance of the system, however, are more noteworthy. Consider a possible new formulation of Mapping3:

[Mapping3.1] $pie{:}Pie(p) \wedge pie{:}name(p,n) \Rightarrow pa{:}Pie(p) \wedge pa{:}pieName(p,n)$

[Mapping3.2] $del:Shop(s) \wedge del:shopName(s,n) \Rightarrow pa:Shop(s) \wedge pa:shopName(s,n)$

[Mapping3.3] $pa:Shop(s) \wedge pa:shopName(s,"Dave's\ Desserts") \wedge shop:Dessert(d) \wedge pa:Pie(p) \wedge pa:pieName(p,n) \wedge shop:name(d,n) \Rightarrow pa:sellsPie(s,p)$

[Mapping3.4] $del:Order(o) \wedge del:orderedBy(o,c) \wedge del:Customer(c) \wedge del:productOrdered(o,pr) \wedge del:Product(pr) \wedge del:productName(pr,n) \wedge del:orderedFrom(o,s) \wedge pa:Shop(s) \wedge pa:sellsPie(s,p) \wedge pa:Pie(p) \wedge pa:pieName(p,n) \Rightarrow pa:orderedPie(c,p)$

[Mapping3.5] $pa:Pie(p) \wedge pa:orderedPie(c,p) \wedge pie:containsFruit(p,f) \Rightarrow pa:likes(c,f)$

This is only one such possible formulation. These restrictions on the mappings result in new, different mappings that are naturally more adaptable to changes in the system. For instance, if Pete's Pies changes their schema, only M3.1 and M3.5 require changes, regardless of how many retail shops or delivery services exist.

One minor disadvantage of this technique is that some intermediate predicates that may otherwise not have been needed in the domain ontology are required. For instance, if the pie aficionados were only concerned with which people liked which fruits, and not with which shops sold which pies, the predicate *pa:sellsPie* would be unnecessary from the perspective of the domain ontology design, but necessary for the mapping sets to be successfully divided. In practice, however, this type of intermediate predicate could be filtered from the results and disallowed in queries, resulting in it effectively not being in the visible domain ontology.

Independent Domain Ontology

In order to have a system in which the users' perspective can evolve over time, it is necessary to have a domain ontology whose definition is independent of that of the data sources. Fortunately, this is enabled almost as a side effect of the independence of the data source mappings outlined above. If one defines the domain ontology before any of the mapping from the data sources is started, and each data source maps into that ontology independently of the others, then a change to the domain ontology is guaranteed to affect the smallest possible number of mappings.

Negation

Finally, we address a problem that is unique to systems based on Semantic Web technologies. As noted before, the majority of these technologies are based on the open world assumption and monotonic reasoning. As such, there is no place for deriving conclusions from the absence of statements. However, other types of sources that one might typically want to integrate operate on a closed world assumption, and thus the lack of a statement may in fact be meaningful.

We address this issue by allowing a restricted form of negation in the rules.

[1] A predicate p may be negated in the antecedent of a mapping $m \in M_d$ if and only if $p \in D_d$

[2] No predicates may be negated in the consequent of a mapping.

This is a reasonable extension because these predicates can only be leaf nodes in the unfolding of the original query, and SPARQL is capable of expressing negation through a combination of an "OPTIONAL" clause and the use of the filter "bound". More formally, for any predicate p that is in need of negation, we can define a predicate p' to mean the absence of p. Since SPARQL allows querying for negation, when the system would query for p', the system can replace it with the SPARQL construction for $\neg p$. Thus this restriction of negation does not add any complexity to the query rewriting process. However, this does imply that our mappings are outside of SWRL as it is currently defined. Since it is yet to be seen whether and in what form negation will be supported in whatever Semantic Web rule language is created by the W3C Rule Interchange Format Working Group [RIF], it seems reasonable to incorporate it conceptually into an information integration system.

The result is negation of predicates in the underlying sources is supported, but the monotonicity of the domain ontology is preserved.

Conclusions

Based on our work, it is clear that previous research on information integration approaches is entirely applicable to an information integration system based on Semantic Web technologies when data source to domain ontology mappings are represented in SWRL. This is because SWRL is a subset of datalog, which is referenced in much of the information integration literature.

Further, we have shown that we can use a modification of the GAV approach to model our information integration system in a way that preserves the stated goals: an independent domain ontology, mutually independent data source to domain ontology mappings, retention of all meaning from data source to domain ontology, and a natural mapping between data source and domain ontologies. We accomplished these goals by starting with the GAV approach and disallowing predicates from multiple data sources in the mappings' antecedents and instead allowing both data source and domain ontology concepts to appear there. This provided both mutually independent data source mappings and a more independent domain ontology. Finally, we added a limited form of negation to the mapping rules, allowing the retention of meaning based on the absence of statements in the data sources. The result is a sound method of utilizing SWRL

in place of relational view definitions in support of a Semantic Web based information integration system.

Future Work

Our continued work in this area is divided along two distinct paths. First, we hope to expand the approach and algorithm to allow for more abstract concept definitions. Second, we have built and are enhancing a prototype system based on the design discussed above.

We hope to extend the approach to make better use of the OWL DL constructs in the domain and data source ontologies. Since many OWL DL reasoners are based on reducing equivalent OWL axioms into LP rules, it seems appropriate to extend the query decomposition reasoning in this direction. Naturally, not all OWL axioms can be expressed in LP [GHVD03], but utilizing those that are is a reasonable starting point. By adding these rules to the query rewriting process, we could begin querying the information integration system with DL constructs, e.g. class intersection and disjunction, etc.

We are also currently working on implementing the three pieces of software SQD, SBRD, and SBWS. This should lead to a proof of concept system which will prove the validity of the approach in the context of a real world example.

References

Calì, A., Calvanese, D., De Giacomo, G., and Lenzenrini, M. 2002. *On the Expressive Power of Data Integration Systems.* London, UK: Springer-Verlag.

Levy, A. 2000. *Logic-Based Techniques in Data Integration.* Norwell, MA: Kluwer Academic Publishers.

Ullman, J. 1997. *Information Integration using Logical Views.* London, UK: Springer.

Levy, A., Rajaraman, A., and Ordille, J. 1996. Querying Heterogeneous Information Sources Using Source Descriptions. In *Proceedings of the Twenty-second International Conference on Very Large Databases.* Bombay, India: VLDB Endowment, Saratoga, California.

Friedman, M., Levy, A., and Millstein, T., 1999. *Navigational Plans for data integration.* In *Proc. Of the 16th Nat. Conf. on Artificial Intelligence,* p 67-73. AAAI Press/The MIT Press.

Xu, L., and Embley, D. *Combining the Best of Global-as-View and Local-as-View for Data Integration. citeseer.ist.psu.edu/604666.html*

Bizer, C., and Seaborne, A., 2004. D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. In *Proceedings of the 3rd International Semantic Web Conference.*Springer.

Dimitrov, D., Heflin, J., Qasem, A., and Wang, N. 2006. Information Integration Via an End-to-End Distributed Semantic Web System. In *Proceedings of the 5th International Semantic Web Conference.* Springer.

Klyne, G., and Carroll, J., eds. 2004. Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/

Dean, M., and Schreiber, G., eds. 2004. OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/

Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., and Dean, M. 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. http://www.w3.org/Submission/SWRL/

W3C Rule Interchange Format Working Group. http://www.w3.org/2005/rules/

Prud'hommeaux, E., Seaborne, A. 2006. SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/

Grosof, B., Horrocks, I., Volz, R., and Decker, S. 2003. Combining Logic Programs with Description Logic. In *Proceedings of 12th International Conference on the World Wide Web.* New York, NY: ACM Press.

Appendix: Example System Definitions

Following are the definitions of the data sources and domain ontologies referenced above, showing classes, properties and cardinalities.

Pie Supplier: (pie:)

- Pie(name, containsFruit*)

Dessert Shop: (shop:)

- Dessert(name)

Delivery Service: (del:)

- Shop(shopName)
- Order(orderedBy, productOrdered*,orderedFrom)
- Customer()
- Product(productName)

Domain Ontology: (pa:)

- Pie(pieName)
- ApplePie(pieName)
- Shop(shopName, sellsPie*, stocksPieWithFruit*)
- Person(orderedPie*,likes*)