Using Large Ontologies to Enable Semantic Interoperability of Problem Solvers

Adam Pease
Teknowledge
1810 Embarcadero Rd
Palo Alto, CA, 94303
apease@teknowledge.com

Abstract

A key to delivering a scalable solution for the integration of knowledge based problem solvers is the notion of semantic integration. We show how semantic integration using a very large knowledge base provides particular leverage in achieving a robust solution. We describe some existing approaches to this problem and then detail our solution applied to several knowledge based problem solvers. We conclude with lessons learned and directions for future research.

Introduction

In this paper, we will explain the notion of senatic integration. We will show this in the context of our work on the DARPA High Performance Knowledge Bases (HPKB) program. The goals of HPKB are to advance the state of the art in knowledge based problem solving, knowledge representation, reasoning and knowledge capture.

The HPKB program participants are constituted into three groups: Challenge Problem Developers, Integrators and Technology Developers. The Challenge Problem Developers are charged with creating test problems which address tasks which are relevant to DARPA's military customers and are amendable to a knowledge based solution. The Challenge Problems (CPs) must balance several objectives including creating a problem which is solvable and yet challenging enough such that it forces the application and development of new technologies. Integrators are charged with solving the CPs and applying whatever technologies are needed to accomplish that task. Technology developers are charged with supplying the integrators with the component technology solutions.

In our role as an integrator we are tasked in part with creating an integrated architecture which allows domain specific problem solvers to work together and contribute to an overall knowledge based solution. Our basis for this integration is the use of a single very large knowledge base. In this first year of the program we have successfully integrated several problem solvers using the Cyc system (Lenat, 1995; Lenat & Guha, 1990).

Background

Software integration is a difficult problem. Systems that were not designed to work together always have a host of different assumptions.

We will divide integration into three layers of connectivity issues. The first is what we will call transport layer issues. This concerns the mechanisms of how to get the bits from one process or machine to another. Solutions include sockets, RMI and CORBA. Another set of issues are found at what we'll call the syntax layer. This concerns how to convert number formats, "syntactic sugar" or the labels of data. The more challenging task is to deal with what we will call semantic connectivity. The integrator must understand the meaning of each data element. Considerable related work has occurred in the database community on the issue of integrating databases at the semantic level [Wiederhold, 1996].

The current state of the practice in software integration consists largely of interfacing pairs of systems as needed. We term this pairwise integration. It is a problem because pairwise integration doesn't scale up. Unanticipated uses are hard to cover later. Chains of integrated systems evolve at best into stovepipe systems. Each integration is only as general as it needs to be to solve the problem at hand.

Some success has been achieved in low level integration and reuse. Systems which use the same scientific subroutine libraries or graphics packages at least are forced into similar representational choices for low level data. DARPA has also invested in early efforts to create large reuse libraries which can assist in integrating large systems at higher levels (Carrico, 1997). Considerable work has gone into expressing a generic semantics of plans in an object oriented format (Pease & Carrico, 1997; Pease & Carrico, 1997:2). Additional work in applying that generic semantics to domain specific applications is promising (Pease & Albericci, 1998).

The development of ontologies for integrating manufacturing planning applications (Tate, 1998) and workflow (Lee, 1996) have also been ongoing.

Another option for semantic integration is to perform software mediation (Park et al, 1997). This could be seen as a variant on pairwise integration, but because integration

is done by knowledge based means, there is an expression of the explicit semantics of the conversion. This renders the effort more reusable.

Personnel at Kestrel Institute have been successful at defining formal specifications for data and using those theories to integrate formally specified software [Srinivas & Jullig, 1995]. In addition, personnel at Cycorp have successfully applied Cyc to the integration of multiple databases.

The Solution

We used a large general purpose knowledge base (Cyc) to link problem solver input and output to the semantic concepts the input and output denote. We augmented the knowledge base with domain or problem specific knowledge to support each specific problem solver. We used knowledge base axioms to create a declarative specification of how problem solver concepts map onto the knowledge base. This approach will now be described for several problem solvers.

Network Flow Problem Solver

One of the HPKB challenge problems can be broadly characterized as providing a battlefield commander with a knowledge level analysis of the characteristics of the battlefield. One element of this is determining how and to what extent goods and personnel can be transported from one location to another. To solve this problem we integrated a Network Flow Problem Solver (NFPS) supplied by personnel at Kestrel Institute.

In the first application of the problem solver we needed to supply a set of roads and the cities they connect and the carrying capacities of the roads in number of vehicles per hour. Note that in this example we were not concerned with sophisticated traffic flow estimation which includes local factors such as temporary backups and congestion. After processing, we wanted the output to be a list of the actual capacities of each road given that each exists as part of a total system. Figure 1 shows an example with capacities on each link. A is given as an infinite source and E is an infinite sink.

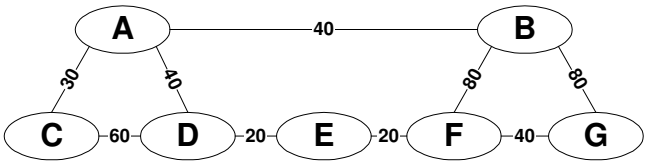

Figure 1 - Network Capacities

Figure 2 shows the same network with the situated capacities on each link. This graph reflects the fact that the only links to node E are bottlenecks. Therefore, even though link B-F can carry 80 units per time period, the actual capacity is limited by the link F-E.

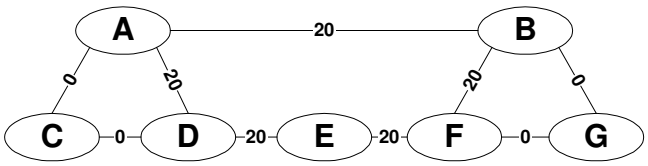

Figure 2 - Situated Capacities

Note that the NFPS has no knowledge of roads or vehicles, only abstract notions of nodes, arcs, capacities and situated capacities. Our next step was to link these abstract notions with concrete notions of roads and vehicles.

Cycorp was already in the process of creating a detailed abstract theory of networks and had already created concepts for roads, vehicles and the flow of traffic. A crucial concept which was not present at first was the notion of a situated capacity as described above..

Our next step in integration was to create a system executive and GUI which allowed user to enter and visualize road information, transformed the road information into input for the NFPS, retrieved the output from the NFPS and then made assertions to the Cyc KB based on that output.

Figure 3 shows the output of NFPS after processing by the system executive. The statements are sent to Cyc. Ellipses denote repetition of the same type of statement for all the nodes or arcs in the network. #$isa is the Cyc instance-of relationship in which an individual is declared to be a member of a class.

```
(#$isa  #$Network1
        #$NetworkFlowSystem-Bounded)
(#$flowTypeOfSystem  #$Network1
                     #$carPerHour)
(#$isa #$Arc1 #$Path-Simple)
…
(#$linkFromToInSystem  #$Arc1
                       #$Node1
                       #$Node2
                       #$Network1)
…
(#$equals   (#$LocalLinkFlowCapacityFn
                 #$Arc1 #$Network1)
            40))
…
(#$equals   (#$SituatedLinkFlowCapacityFn
                 #$Arc1 #$Network1)
            20)
…
```

Figure 3 - NFPS output to Cyc

A block diagram of the entire system is shown in Figure 4.

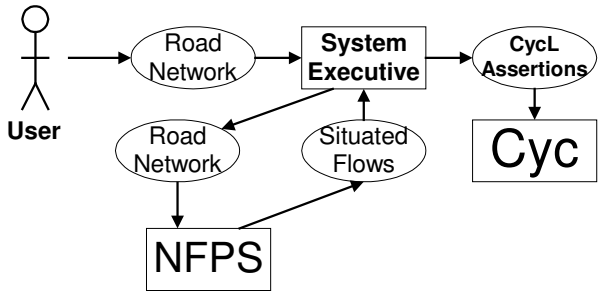

Figure 4 - Network Flow Problem Solver block diagram

Dynamical Recognizers

The central thesis of this technology development task is that representations of dynamics are the foundation for knowledge level coding of verbs (Cohen, 1998). While the first iterations of the challenge problem do not truly stress this technology, an initial integration provided practical feedback on its eventual application as part of a sophisticated knowledge based system.

The battlespace challenge problem provides a large data set describing the movements of battlefield entities over a several hour period. The first level of understanding the significance of the data is to classify the vehicles and battlefield sites based the movement data. We integrated several recognizers that use statistical pattern matching to identify patterns in the data. Each recognizer identifies one type of physical battlefield location. Two of these locations are refueling stations and command posts.

The statements in Figure 5 assert the location of a particular site and the time at which its existance began.

```
(#$latitude  #$Refueling34
             (#$Degrees 54))
(#$longitude #$Refueling34
             (#$Degrees 25))
(#$after
     (#$StartFn #$Refueling34
                (#$DayFn 14
                  (#$MonthFn #$February
                    (#$YearFn 1996)))))
```

Figure 5 - Sample Site Recognizer Assertions

Workarounds Inputs

The workarounds challenge problem is another part of the overall battlespace problem for HPKB. It addresses the task of finding engineering workarounds for a disrupted transportation network. When confronted with a damaged bridge or road, military engineers must find repair solutions which address the capabilities of the vehicles which need to pass, the available engineering assets, and the characteristics of the terrain and damaged transportation infrastructure.

This problem was posed by the Challenge Problem Developers as a database which characterizes the damaged infrastructure, engineering assets and terrain. Our integration task was to transform that database into a set of assertions to a knowledge base and to link that to knowledge based workaround reasoner which is built in Cyc. That reasoner then performs workarounds problem solving to recommend a solution.

The database has several records which are utilized in specifying the current problem (from [Jones, 1998]):

1. Unit. A military unit with its nominal and actual properties
2. Vehicle-Of. Assigns vehicle counts to Units (by type of vehicle)
3. Equipment-Of. Assigns equipment counts to Units (by equipment type & vehicle type)
4. Unit-Of. Assigns specific Units as parts of other Unit
5. Site. A geographically located militarily-relevant place
6. Site-Point. A point that is part of the definition of site geometry
7. Site-Part. Assigns specific Features/Sub-Sites to some Site
8. Site-Attrib. Free-form attributes for Sites
9. Damage. Specific degradation of some Site/Feature
10. Damage-Attrib. Free-form attributes for Feature-Damage

The workarounds problem solver first maps these database records into concepts in the knowledge base. It then poses hypothetical actions to Cyc for a determination of whether the actors in the problem are capable of the actions. Actions may need to be decomposed or enabled by the results of other actions. A block diagram of the system is given in Figure 6.

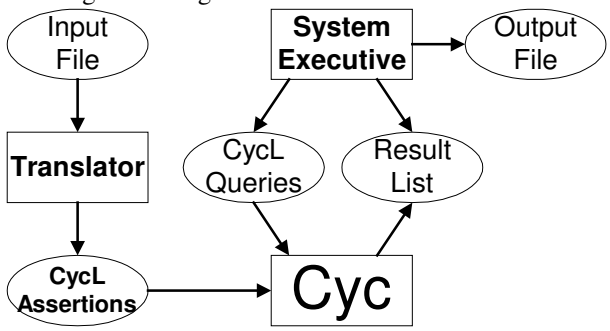

Figure 6 - Workarounds Problem Solver

Future Work

The real payoff in semantic integration will come in later phases of this project when each of the systems described above are linked into a single tool suite for battlespace problem solving. Before this paper is in print we will have received several more products to be integrated. Among

them are tools for terrain analysis, trafficability analysis and route finding. We will also be integrating a commercial Geographic Information System (GIS).

The terrain analysis tool will examine terrain information supplied by the GIS and produce a set of polygonal regions which are labeled as to their characteristics relative to movement actions. This analysis is independent of vehicle type or other factors in the situation.

The trafficability analysis tool will relate the results of the terrain analysis tool to the specific situation given in the challenge problem. It will provide estimates of movement costs both in time and resources used for specific vehicles, combat and weather conditions.

The route finding tool will work with the outputs from the trafficability tool. It looks at the list of movement costs as an abstract graph and finds a set of the best paths through the network with regards to those costs. Those paths can then be added to the transportation network which is the input to the Network Flow Problem Solver described previously.

While the flow of information for these tools is relatively sequential, the outputs of each are relevant for the workarounds problem solver. By virtue of having integrated each tool with the Cyc KB, there will be no need to have a separate integration with the workarounds problem solver. The integration will already have been done.

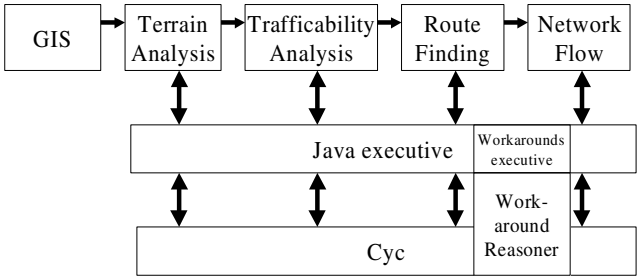

Figure 7 - Full Battlespace System

Conclusion

In this paper we have attempted to show the process of achieving semantic integration by describing the integration of several problem solvers. We have shown how these problem solvers are specialized for a particular problem. We have alluded to the benefits provided by our integration approach over a more conventional pairwise integration.

Acknowledgements

Thanks first to Doug Lenat for his years of work in leading the development of Cyc, without which, our integration effort would not be possible. Thanks to Fritz Lehmann and Ming Xu at Cycorp for creating the network axioms used with NFPS, to Cleo Condoravdi and John Li at Teknowledge for their excellent work on integration, Paul Cohen and his group at UMass for their help in integrating the dynamical recognizers, and Lee Blaine and John Anton at Kestrel for their help in integrating NFPS. Thanks also to Dave Gunning, the DARPA HPKB program manager for creating and supporting the program which made this work possible.

References

Carrico, T., (1997) Object Model Working Group, Command and Control Schema, Revised Draft, Version 0.5.3. Unpublished report.

Cohen, Paul R. (1998). Dynamic Maps as Representations of Verbs. Submitted to ECAI-98.

Jones, E., (1998), "Formalized Intelligence Report Ensemble (FIRE) & Intelligence Stream Encoding (ISE)", Alphatech memo.

Lee et al, (1996) The PIF Process Interchange and Framework Version 1.1, Jintae Lee (editor) Micheal Grunninger, Yan Jin, Thomas Malone, Austin Tate, Gregg Yost and other members of the PIF Working Group, published as MIT Center for Coordination Science, Working Paper #194, 1996. Available via http://soa.cba.hawaii.edu/pif/

Lenat, D. and Guha R., (1990) Building Large Knowledge Based Systems. Reading, Massachusetts: Addison Wesley.

Lenat, D. (1995) Artificial Intelligence. *Scientific American*, September.

Park, J., Gennari, J., & Musen, M., (1997). Mappings for Reuse in Knowledge-based Systems. Stanford Section on Medical Informatics technical report SMI-97-0697.

Pease, A., & Albericci, D., (1998). The Warplan, Teknowledge, Palo Alto, CA, March 13.

Pease, A. & Carrico, T. (1997). "The JTF ATD Core Plan Representation: Request for Comment". Armstrong Lab: AL/HR-TP-96-9631.

Pease, A. & Carrico, T. (1997). "The JTF ATD Core Plan Representation: A Progress Report", Proceedings of the AAAI 1997 Spring Symposium on Ontological Engineering.

Srinivas, Y. & Jullig, R., (1995) Specware(TM): Formal Support for Composing Software, Proceedings of the Conference on Mathematics of Program Construction, Kloster Irsee, Germany, July.

Tate, A. (1998), "Roots of SPAR - Shared Planning and Activity Representation", to appear in Knowledge Engineering Review, Special Issue on Ontologies, Cambridge University Press.

Wiederhold, G. (ed.) (1996): Intelligent Integration of Information; Kluwer Academic Publishers, Boston MA, July 1996.