TheaterLoc: Using Information Integration Technology to Rapidly Build Virtual Applications

Greg Barish, Yi-Shin Chen, Dan DiPasquo, Craig A. Knoblock,
Steven Minton, Ion Muslea, Cyrus Shahabi
Information Sciences Institute, Integrated Media Systems Center and
Department of Computer Science, University of Southern California
{barish, dipasquo, knoblock, minton, muslea}@isi.edu, {yishinc, shahabi}@usc.edu

Abstract

Although there has been much written about various information integration technologies, little has been said regarding how to combine these technologies together to build an entire application. We demonstrate TheaterLoc, an information integration application that allows users to retrieve information about theaters and restaurants for various U.S. cities, including an interactive map depicting their relative locations. The data retrieved by TheaterLoc comes from five distinct heterogeneous and distributed sources. The enabling technology used to achieve the integration includes: the Ariadne information mediator, a web site wrapper learning tool, the Theseus execution system, and a mechanism for distributed spatial query planning. Our system is novel because it demonstrates how "virtual applications" can be rapidly built from a set of integration tools and existing online data sources.

1. Introduction

TheaterLoc [1] is an application that integrates multiple, heterogeneous data sources in order to gather information about theaters and restaurants for various cities in the United States. The application is built using a set of tools: an information mediator, a web site wrapper learner, a mechanism for distributed spatial querying, and a dataflow-based plan execution system. TheaterLoc is an example of what is involved in using these tools to build an information integration application.

Our demonstration consists of three parts: (a) running the standard TheaterLoc application to show the data integration, (b) using a GUI to show how Web sites can be *wrapped*, and (c) using an additional interface to show how spatial data querying is supported by the system.

2. The TheaterLoc Application

TheaterLoc can be accessed on the Internet using any Web browser. Users choose which a city for which they would like to locate theaters and restaurants. The system returns this data in tabular format, as well as an interactive map identifying the relative locations of each restaurant and theater within that city (Figure 1). Users can click on any of the map points to be taken to a web page containing further details about that particular place. When choosing a theater, users also have the option of viewing a listing of movie showtimes, along with video trailers for some films.

The information retrieved by TheaterLoc comes from five online sources. Restaurant data is gathered from CuisineNet, theater information from Yahoo, and the video trailers from Film.com. Construction of the interactive map is facilitated by two sources: the ETAK Geocoder (for plotting points on a map) and the US Census Tiger Map Server. The TheaterLoc application is effective because it seamlessly integrates the useful data from these sources, automatically correlating them as necessary.

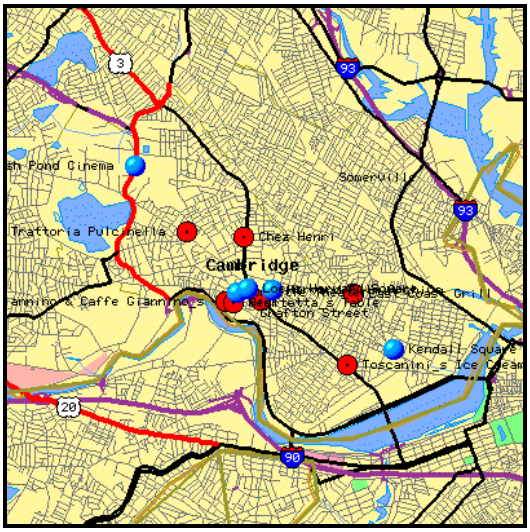

Figure 1: Restaurants and theaters in Cambridge, MA

3. The Construction of TheaterLoc

TheaterLoc is built using a set of data integration tools. The Ariadne information mediator is used to perform run-time integration of heterogeneous data sources. Wrappers are used as proxies to query web sites for information. A wrapper learning tool is used to automatically generate wrappers through demonstration. A dataflow-based plan

execution system is used to run the plan generated by the mediator. Finally, an alternative spatial query interface is used to implement distributed spatial querying.

Both the information mediator [4] (which combines techniques related to domain modeling, axiom pre-compilation, and query planning) and dataflow plan execution system [5] are demonstrated by running the TheaterLoc application through its standard interface. In addition, we are demonstrating the wrapper learning tool and distributed spatial querying.

3.1. Wrapper Learning

For Ariadne and Theseus, *wrappers* provide a way to download web pages and return data as a logical relation, for use in subsequent relational operations (join, project, etc.). Embedded in the wrappers is a technology for information extraction, which we call STALKER.

The STALKER system consists of both a learner and an extractor. The system works using embedded catalogs (ECTs) and extraction rules. And ECT specifies the logical hierarchical structure of data on a web page. The extraction rules describe how to parse the web page in order to extract various elements identified in that catalog.

The STALKER learning component allows extraction rules to be generated from the simple "marking up" of that web page by a user, using a Wrapper Learner GUI tool. The mark-up process only requires that the user demonstrate a few examples of the various attributes contained in the web page. Learning techniques are then used to deduce the extraction rules. The STALKER executor is a run-time mechanism for extracting data based on the contents of the ECT and extraction rules.

Our demonstration includes showing how the Wrapper Learner GUI is used to build STALKER rules and ECTs through a user mark-up process. Thus, we will show how this tool can be used to automatically generate stand-alone wrappers by user-demonstration.

3.2. Spatial Data Querying

We demonstrate distributed spatial querying using an alternative interface to TheaterLoc. This interface queries both Ariadne and a local database with spatial query capabilities. Currently, we support two alternative local database servers: the Informix Universal Server (with its Geodetic datablade) and the NCR Teradata Object Relational server. Through this interface, we use TheaterLoc to support queries such as *show restaurants and theaters within a 10 km radius of a street address*. This interface is slightly different than the standard TheaterLoc interface, as it is based on restaurants and theaters obeying a radius, not within a specific city.

A given spatial query is executed in three steps. First, a local spatial select is performed to obtain the city polygons *overlapping* with the user-defined circle. Second, the names of these cities are used to fetch the corresponding theater and restaurant coordinates from Ariadne. Finally, another local spatial select filters out those coordinates that are not *contained* within the user-defined circle.

These distributed spatial queries introduce new challenges in the area of spatial query optimization. Previous work on distributed spatial query optimization assumed full control of information sources, including some that also assume the existence of spatial index structures on both relations. Under those assumptions, the most efficient way of supporting join queries was to lay an identical grid on each of the participating sites, thereby reducing the spatial join to a regular join between the unique identifiers of the grid cells.

However, with TheaterLoc, we can only support restricted versions of *spatial selects* on Ariadne and cannot assume the existence of any sort of spatial index structures such as R-trees or grids. In addition, we do not have write access on the web-sources or Ariadne. Currently, we are investigating alternative execution plans for such queries. We demonstrate the execution of three different plans for a given query, each plan making different compromises between response time and accuracy.

4. Discussion

The Internet is maturing into an essential medium for information distribution. Information integration is becoming an important technology for extracting and correlating data from Web so that it can be used to build powerful applications. In this demonstration, we present TheaterLoc, which provides an example of how such applications can be constructed. We show that by combining an information mediator with a wrapper learning tool, a dataflow plan execution system, and a spatial query mechanism, virtual applications can be rapidly built.

5. References

[1] Barish, G; Knoblock, C.A.; Chen, Y-S; Minton, S.; Philpot, A.; Shahabi, C. TheaterLoc: A Case Study of Information Integration. IJCAI Workshop on Information Integration, Stockholm, Sweden. 1999.

[2] Ambite, J.L. and Knoblock, C.A. Planning by Rewriting: Efficiently Generating High-Quality Plans. *Proc of the 14th Natl Conf on Artificial Intelligence*, Providence, RI. 1997

[3] Muslea, I.; Minton, S.; and Knoblock, C.A. A Hierarchical Approach to Wrapper Induction. *Third Conference on Autonomous Agents*, Seattle, WA. 1999

[4] Knoblock, C.A.; Minton, S; Ambite, J.L.; Ashish, N.; Modi, J.; Muslea, I.; Philpot, A. and Tejada, S. Modeling Web Sources for Information Integration. *Proc of the 15th Natl Conf on Artificial Intelligence*, Madison, WI. 1998

[5] Barish, G.; DiPasquo, D.; Knoblock, C.A.; Minton, S. Efficient Execution for Information Management Agents. *ACM CIKM Workshop on Web Information and Data Management.* Kansas City, MO, USA. 1999.