A Mixed-Initiative System for Building Mixed-Initiative Systems

Craig A. Knoblock, Pedro Szekely, and **Rattapoom Tuchinda**
Information Science Institute
University Of Southern California
4767 Admiralty Way
Marina del Rey, CA 90292

Abstract

Mixed-initiative assistants can be applied to a variety of information-rich problem-solving tasks on the Web, such as travel planning and equipment purchasing tasks. A mixed-initiative environment for such tasks can greatly improve the decision making environment for a user if the application is designed to meet the needs of a user. However, each user has different needs and preferences, making it difficult to design a single application for all users. Thus, we are applying the mixed-initiative paradigm recursively to develop a mixed-initiative system for building mixed-initiative systems. This paper describes the basic framework for constructing mixed-initiative systems, which is based on our previous work on developing mixed-initiative information assistants in Heracles. The new system, called Alcmene, will be implemented as an application of Heracles and will allow a user to author a new Heracles application through a mixed-initiative problem-solving process.

I. Introduction

In previous work, we developed a constraint-integration framework, called Heracles, which was designed for implementing mixed-initiative, multi-source information assistants (Knoblock *et al.* 2001a). This system has been applied to a variety of useful applications, including travel planning (Ambite *et al.* 2002), visitor scheduling (Chalupsky *et al.* 2001), and geospatial data integration (Knoblock *et al.* 2001b). However, each of these applications of Heracles required a significant effort to build and each successful application invariable generated requests for changes, additional sources, and new features. Ideally, we would like to allow the users of the system to create, update, and improve their own mixed-initiative applications.

In this paper we describe an approach to building new applications and modifying existing applications using a mixed-initiative approach. The basic idea is to construct a mixed-initiative application in Heracles that interactively constructs mixed-initiative applications. Heracles is used for both the authoring environment as well as the resulting application, so the user would see a single uniform interface for the entire system. In order to support the authoring, we address the issues of how to identify the relevant sources, how to link the sources together, and how to relate the specific data instances from the various sources.

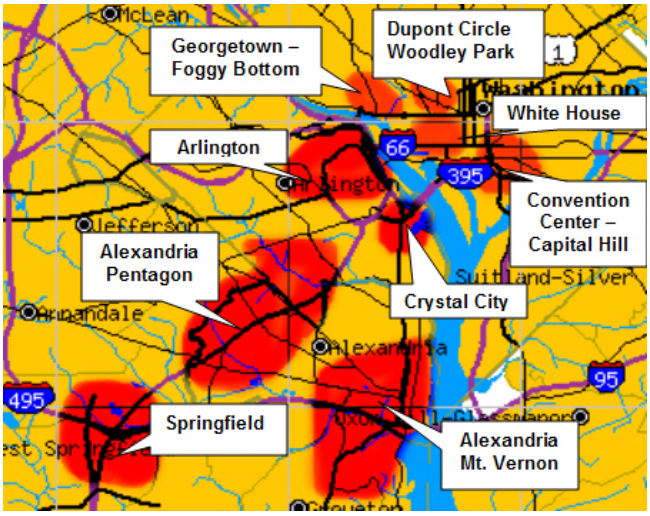

Figure 1: Area segmentation on Priceline for Washington DC area

The paper is organized as follows: Section 2 describes a motivating example that is then used throughout the rest of the paper. Section 3 describes how this example would be solved by defining a Heracles application. Section 4 describes the issues in creating an authoring environment for constructing such an example in Heracles. Section 5 compares the work presented here to previous work on user interfaces and mixed-initiative systems. And section 6 summarized the contributions and future directions of this work.

II. Motivating Example

When travelling to an unfamiliar destination, one often does research on the surrounding area of interest. The decision of where to stay differs based on each person's preferences. For example, assume Mary needs a hotel near a meeting location at 7701 Telegraph Road, Alexandria, VA, 22315. To save money, she wants to use Priceline.com to book a room. Figure 1 shows how Priceline divides the area in Virginia. On Priceline, there are three areas (Springfield, Alexandria-

4* Marriott J.W. Hotel White House, 7/15-18, $80	0	7/13/05 12:48 am	al88
4* Dupont Circle - Omni Shoreham 7/18-20 $85	0	7/9/05 10:17 pm	cjhorh
2.5* Residence Inn Vermont Ave 7/1-7/3 $47; 7/3-7/5 $55	0	7/9/05 7:35 pm	kirklt
3* Mariott Washington Westend 6/28/05 $75	0	7/9/05 4:37 pm	kirklt
3* Renaissance Washington DC Hotel 6/29/05 to 7/1/05 $62	0	7/9/05 4:32 pm	kirklt
3* Holiday Inn Hotel and Suites Alexandria 6/27/05 $52	0	7/9/05 4:30 pm	kirklt

Figure 2: A post on BiddingForTravel.com website includes winning price, area, hotel, star rating, and reservation date

Pentagon and Alexandria-Mt. Vernon) among nine areas that are close to the meeting location. It would be convenient to secure a room in any of those three areas. However, if the price is too expensive or none of the three or four star hotel is available in those area, Mary is willing to stay in other areas and opt to drive instead.

To get a sense of the price range, she relies on BiddingForTravel.com which provides names of the hotel used by Priceline and historical data of successful bids on Priceline for hotels in each area. Figure 2 shows how the bidding information is posted on the site. For example, the first row reports that on 7/13/05 a customer successfully bid $80 and got the Marriott hotel. She assumes that similar bids will be accepted on her travel days.

Armed with the list of hotels obtained from BiddingForTravel, Mary goes to Orbitz to find the lowest price (in case there exist better choices for similar price) and to find the address of each of the hotels. Once she has all the hotel addresses, she can geocode them along with the meeting address and display those locations on the map.

At this point, Mary will have enough information to decide whether she should book one of the hotels through Orbitz or make a bid on Priceline for a hotel in the area.

III. Heracles

In previous work, we developed a system for mixed-initiative planning called Heracles (Knoblock *et al.* 2001a; Michalowski *et al.* 2004). Heracles supports interactive planning in information-rich application domains. It provides a constraint-integration framework that dynamically retrieves and integrates information from multiple sources. This framework exploits constraint-propagation techniques to combine information gathering, integration, and display in a single unified system. Heracles has been applied to a number of application areas including travel planning, travel monitoring, and geospatial information gathering.

Consider the application of Heracles to the motivating example described in the previous section. As shown in Figure 3, the user would first enter the address, city, state, zipcode, dates, star rating, and Priceline area in a top-level template in Heracles. Each input value provides a setting for a

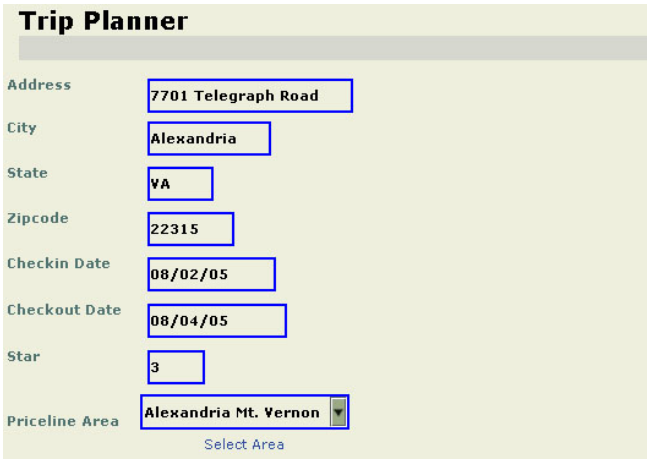

Figure 3: The top level template where users specify trip information

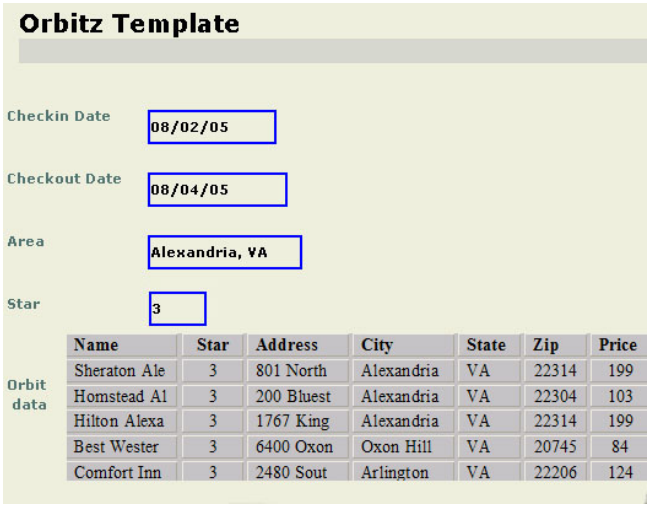

Figure 4: Variables from the parent template can propagate down and provide input to retrieve hotel data from Orbitz

Figure 5: The final template where Orbitz, BiddingForTravel, and map information are integrated and overlaid on a satellite image of the area

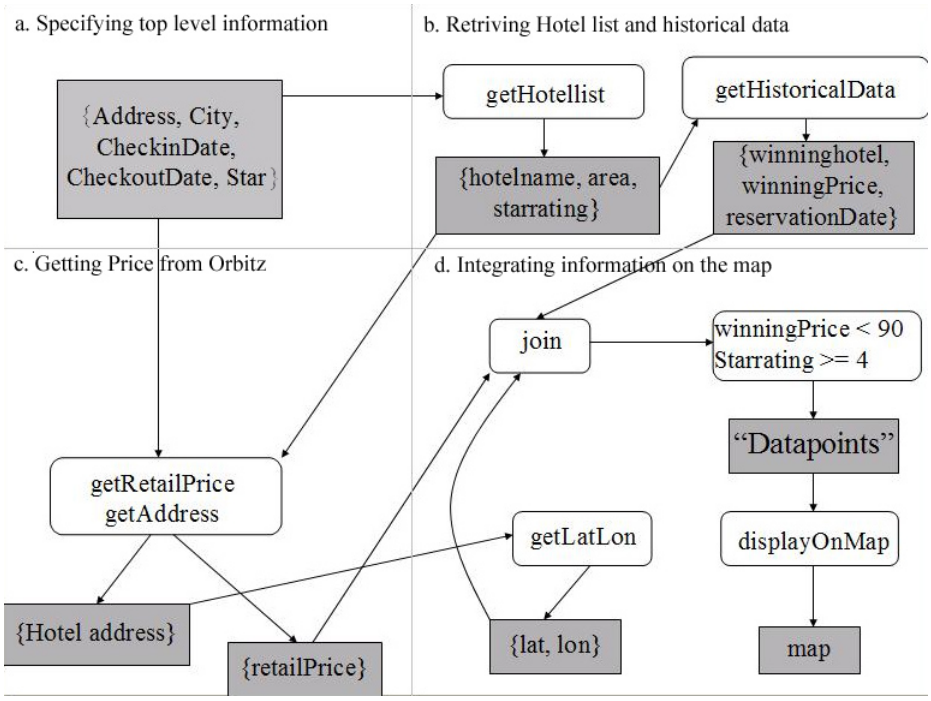

Figure 6: Schematic representation of the Heracles templates for the scenario. The overall plan can be divided into 4 templates. Template A, B, and C retrieve data from different sources and the information is integrated in template D where the results are shown on the map

variable in the constraint network and the effect of setting these values triggers constraints that perform additional information gathering and integration operations. In this case, the inputs for Priceline area and star rating would trigger a call to the BiddingForTravel web site to retrieve the set of Priceline hotels for the given area at that star rating. At the same time, the city and state would trigger a call to the Orbitz web site to retrieve the hotels in that area. Once both datasets have been retrieved, the results are then combined and shown on a new page (Figure 4), which provides the current prices from Orbitz for the Priceline hotels that are listed on the BiddingForTravel web site.

Given the list of hotels and their addresses, the Heracles system would then call a geocoding source, which maps the hotel addresses into lat/long coordinates. The system would also look up the reported successful bids in BiddingForTravel for each of the hotels in the desired area at the specified rating. Next, the system would then retrieve a satellite image of the area from an online source and display the set of hotels, their Orbitz price, and the reported successful bids. This final result is shown in Figure 5. The hotels are shown with a dot on the image and the user can then mouse over each dot to get the information specific to that hotel. The image also shows the location of the hotels relative to the original address. Using this application of Heracles, a user can interactively explore the different Priceline areas and hotel ratings to select a hotel to reserve in Orbitz or to make a bid on a hotel area in Priceline.

While this type of Heracles application is very useful, there is a significant amount of effort required to build such an application. An author of such an application must identify an appropriate set of data sources, specify how the sources are queried and linked, define the variables and constraints for the constraint network, and design the overall layout of the information. Once the application is built, it can be used over and over, but the construction of such an application in Heracles today requires detailed knowledge of the constraint language and structures within Heracles. No casual author would want to program templates because the time required would outweigh the benefit. This leads us to the idea of creating a mixed-initiative authoring environment in Heracles for authoring new applications in Heracles.

IV. Mixed-initiative Authoring

We are developing a mixed-initiative authoring system called Alcmene.[1] Built on top of Heracles, Alcmene allows authors to build new Heracles templates or to modify existing templates with no or minimum coding; anyone who could write simple excel expressions should have no trouble with creating templates. Alcmene provides the author with dynamic Heracles templates that mutate to offers choices and operations based on the current constraint structure.

We use the scenario discussed as an example to illustrate the structure of the problem involving mixed-initiative authoring. Figure 6 shows the schematic detail of the plan for the scenario. Template A allows the author to specify top-

[1] In Greek mythology, Alcmene was the mother of Heracles.

level information as shown in Figure 3. The information from Template A also propagates to Template B (which retrieves hotel list and historical bidding data) and C (which retrieves current pricing and address from Orbitz as shown in Figure 4). Template D integrates all information and put the results on the map as shown in Figure 5.

We assume that the data is extracted from the source using wrappers (Knoblock *et al.* 2003) for semistructure data (Orbitz) and semantic annotation (Michelson & Knoblock 2005) for unstructured data (BiddingForTravel).

In general, we can view the plan building process as the problem of trying to connect data sources together in the right way. This process can involve identifying, integrating, and filtering data sources.

Based on this observation, the top-level goal for our approach is to facilitate the process of connecting data sources together. We must solve three main problems.

- *Identifying the sources to use*

  During the whole authoring process, an author might know some of the sources she wishes to use, but may not know which intermediate sources to use to connect them together. For example, to put the address of a hotel on the map, we need to get its latitude and longitude, but an author might not know that latitude and longitude are required in order to put an address on a map. This problem involves determining how many intermediate sources we should use to form a path to connect two selected sources.

- *Linking the sources together*

  Two sources could be linked together when the required input and output matched. However, authors might decide to put additional constraints between the sources to filter or transform the data. For example, if the author know that the JW Marriott hotel has a hidden 16 dollars parking fee, the author might want to put a constraints that adds 16 dollars to the retail price returned from Orbitz.

- *Linking the data together*

  Data returned from different web sources can have different formats. For example, the date returned from one source might have the format MM-DD-YY, while the date required as input to another source might be MM-DD-YYYY. Given multiple web sources and their dynamic nature, the author might know about the kind of the data (i.e., date) that each source provides, but might not know the exact format.

  The other problem when authors try to link the data together is record linkage. Connecting data from multiple sources together might require that some attribute values be matched. For example, the hotel's name mentioned in BiddingForTravel is *Sheraton Suites Alexandria*, while the same hotel name from Orbitz is *Sheraton Suites Old Town Alexandria*. In Alcmene, we plan to integrate an available record linkage system, such as Apollo (Michalowski, Thakkar, & Knoblock 2004), to specifically address this problem.

We plan to tackle each problem by using the techniques described below.

Source Definition and Bidirectional Search

Finding a set of candidate sources between two existing sources is a search problem. As a result, we first need a way to represent a source. Each source is defined in the form of a predicate specifying its input and output. For example, we can define Orbitz, Geocoder, and SatelliteMap data sources as follow:

Orbitz($indate, $outdate, $city,
hotelname, price, address)
Geocoder($address, lat, lon)
SatelliteMap($lat, $lon, $data, image)

Each source is defined in terms of what is required as input and output; the $ sign in front of an attribute means that the attribute is required as input.

Once we have each source definition, we can identify all the paths consisting of one or more sources between two selected sources. For example, if an author already has Orbitz and SatelliteMap as selected sources, we could execute a bidirectional search through data source predicates to identify how many paths exist between Orbitz and SatelliteMap. Then the author can choose the path that suits his needs.

Immediate Partial Plan Execution

We choose to show the result to the authors in every execution step, which allows them to see if the data returned from each connecting point of the plan is what they expect or not. For example, in Template c of Figure 6, the retail price returned by Orbitz might be $95 instead of just a number 95. By seeing the intermediate result, the author can decide to set up a constraint to filter out the dollar sign.

This approach allows the author to see results when adding new constraints and to understand the data format returned by data sources, which helps us solve the problems of linking the sources together and linking the data together.

Constraint Specification

We provide a default set of functions that authors can use to perform simple filtering operations, such as select, project, join, and replace. Authors can also create simple excel-like expressions. For example, the hidden 16 dollars parking fee could be specified as IF(b.winninghotel = 'JW Marriott', b.winningprice = b.winningprice + 16, b.winningprice). Furthermore, an expression in one template can make a reference to variables in another template.

Added to a path between two data sources, constraints allow the author to handle inconsistent data formats (linking data), perform filtering, and specify their own constraints (linking sources).

V. Related Work

Heracles applications are visual applications that integrate and fuse information from multiple sources and render them as a dynamic web page. For example, the hotel application integrates information from Priceline, BiddingForTravel and Orbitz, and renders the data on the image that integrates maps, imagery and vector data sources.

In this respect, Alcmene is similar to tools for authoring dynamic web pages ((Miller & Myers 1997), (Wolber, Su,

& Chiang 2002) and (Macias & Castells 2003)). The main difference between Alcmene and these systems is that Alcmene supports the construction of much more sophisticated applications that combine and fuse the information rather than simply retrieve information from multiple sources and showing the retrieved items side by side. For example, Turquoise (Miller & Myers 1997) can be used to construct a dynamic web page that pulls weather information from one web site and news information from a different site. The results are shown as different sections in the resulting page. In contrast, Figure 5 shows how Alcmene can be used to construct dynamic pages that fuse the information coming from the different sources. The pop-up window showing hotel information combines information from Orbitz.com and BiddingForTravel.com, and the information has been geocoded using another source so that it can be shown on the image.

Alcmene and these systems also differ in terms of the user interface paradigm of the authoring environment. These systems rely heavily on programming by demonstration and programming by example paradigms. Consequently the author edits the interface that the end-user would see or provides examples of the data that the end user would like to see. The system generalizes the editing actions or examples to build general scripts or models of the application. Alcmene is targeted for applications with more complex data transformations and behavior that are difficult to specify by example or demonstration. Instead, Alcmene uses a specification plus preview paradigm similar to that used in popular web development tools, such as Macromedia Dreamweaver.[2] In these tools the author edits the specification and then issues a preview command to see how the application would look to the user. In Alcmene, the two views are always visible to the author, and the preview is automatically updated each time the user edits the specification. This allows the author to immediately see the consequences of changes to the specification, and allows the author to work incrementally. The system is able to preview partial and even incorrect specifications, showing in the preview as much as can be computed.

Our system does not directly address the problem of constructing wrappers for web sites. The system is designed to use any wrapper construction tool. We use the tool from Fetch Technologies,[3] based on our prior work (Muslea, Minton, & Knoblock 2001). This tool does use programming by demonstration and learning techniques as is done in other tools (Miller & Myers 1997).

Heracles applications can also be viewed as mixed-initiative planning applications. Once users provide their initial input, the system performs steps in a plan automatically until it reaches choice points where the user enters information that the system uses to select an appropriate plan expansion. The authoring tool, Alcmene, is itself implemented in Heracles, and consequently can also be viewed as a mixed-initiative planning application. Alcmene uses plan sketching techniques similar to PASSAT (Myers *et al.* 2002) and CAT (Kim, Spraragen, & Gil 2004) where the author can insert plan fragments and the system tries to link those plans with the rest of the plan, allowing the user to combine top-down and bottom-up refinement of the hierarchical plan representation. Because Alcmene is targeted towards simpler plans containing tens rather than hundreds of operators, the techniques used to expand tasks, instantiate variables, and flesh out sketches are different. PASSAT and CAT rely on sophisticated domain models and deeper reasoning to offer advice. Alcmene works with a plan instance (the preview) that allows the author to see the behavior of the plan to detect errors that cannot be detected automatically from the shallower models. The additional advantage of working with a plan instance is that the user can get a better feel for the behavior of the plan and detect misconceptions that lead to unexpected and unwanted results.

[2] http://www.macromedia.com

[3] http://www.fetch.com

VI. Discussion

Alcmene is an authoring tool for Heracles, our infrastructure for building mixed initiative planning applications for constructing dynamic web pages that integrate and fuse information from multiple sources. Alcmene is being built using Heracles given that we view authoring as a mixed initiative planning activity. Both the authoring tool and the applications built using the authoring tool are Heracles applications. This allows us to tightly integrate the authoring tool and the authored application, allowing us to constantly keep the preview up-to-date with the specification.

This approach also allows us to produce extensible applications. Our vision is to blur the distinction between authors and users. We have been careful to use the two terms in this paper to clearly describe the different roles. However, we expect that the same person will take on the different roles at different times. A user will start with an application that closely matches the task that the user wants to do, but may decide to first customize it to fit the task better. For example, we may like the hotel application described in this paper, but would rather use Travelocity instead of Orbitz, and we may also want to see whether the hotels have a Health club because we would like the opportunity to exercise after a busy day of meetings. Because both the application and the authoring tool are built using Heracles, they offer the same user interface paradigm so the interface to the authoring tool will be familiar to the users of the application. Also, users can decide to invoke the authoring tool after they started using the application, once they realize that they want to see an additional piece of information. They can extend the application at that point, keeping all the data they have produced so far, integrating new capabilities in the current context, and then continue using the application.

We are actively working on Alcmene. We have a design for it and are currently working on the implementation. The authoring tool will provides the author the operators to manipulate the specification being built. In addition, it will also provides support to extend and refine the specification, suggesting operators that can be applied to refine the specification.

VII. Acknowledgment

This research is based upon work supported in part by the National Science Foundation under Award No. IIS-0324955, in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010, in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-1-0504, and in part by the Air Force Office of Scientific Research under grant number FA9550-04-1-0105.

The U.S.Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

References

Ambite, J. L.; Barish, G.; Knoblock, C. A.; Muslea, M.; Oh, J.; and Minton, S. 2002. Getting from here to there: Interactive planning and agent execution for optimizing travel. In *Proceedings of the Fourteenth Conference on Innovative Applications of Artificial Intelligence (IAAI-2002)*, 862–869.

Chalupsky, H.; Gil, Y.; Knoblock, C. A.; Lerman, K.; Oh, J.; Pynadath, D. V.; Russ, T. A.; and Tambe, M. 2001. Electric elves: Applying agent technology to support human organizations. In *Proceedings of the Conference on Innovative Applications of Artificial Intelligence*.

Kim, J.; Spraragen, M.; and Gil, Y. 2004. An intelligent assistant for interactive workflow composition. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interface*, 125–131. New York, NY, USA: ACM Press.

Knoblock, C.; Minton, S.; Ambite, J. L.; Muslea, M.; Oh, J.; and Frank, M. 2001a. Mixed-initiative, multi-source information assistants. In *Proceedings of the World Wide Web Conference*.

Knoblock, C. A.; Ambite, J. L.; Minton, S.; Shahabi, C.; Kolahdouzan, M.; Muslea, M.; Oh, J.; and Thakkar, S. 2001b. Integrating the world: The worldinfo assistant. In *Proceedings of the 2001 International Conference on Artificial Intelligence (IC-AI 2001)*.

Knoblock, C.; Lerman, K.; Steven, M.; and Muslea, I. 2003. Accurately and reliably extracting data from the web: A machine learning approach. In *Intelligent Exploration of the Web*, 275–287. Berkeley, CA.

Macias, J., and Castells, P. 2003. Dynamic web page authoring by example using ontology-based domain knowledge. In *Proc. of the Eight International Conference on Intelligent User Interfaces*.

Michalowski, M.; Ambite, J. L.; Knoblock, C.; Minton, S.; Thakkar, S.; and Tuchinda, R. 2004. Retrieving and semantically integrating heterogeneous data from the web. *IEEE Intelligent Systems* 19(3):72–79.

Michalowski, M.; Thakkar, S.; and Knoblock, C. 2004. Exploiting secondary sources for unsupervised record linkage. In *Proc. of 2004 VLDB Workshop on Information Integration on the Web*. Toronto Canada, August 2004.

Michelson, M., and Knoblock, C. 2005. Semantic annotation of unstructured and ungrammatical text. In *Proc. of 19th IJCAI*. Edinburgh, Scotland.

Miller, R., and Myers, B. 1997. Creating dynamic world wide web pages by demonstration. Technical report, Carnegie Mellon University School of Computer Science. CMU-CS-97-131.

Muslea, I.; Minton, S.; and Knoblock, C. A. 2001. Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems* 4(1-2):93–114.

Myers, K. L.; Tyson, W. M.; Wolverton, M. J.; Jarvis, P. A.; Lee, T. J.; and desJardins, M. 2002. Passat: A user-centric planning framework. In *Proceedings of the Third International NASA Workshop on Planning and Scheduling for Space*.

Wolber, D.; Su, Y.; and Chiang, Y.-T. 2002. Designing dynamic web pages in the WYSIWYG interface. In *Proc. of the Sixth IFIP Working Conference on Visual Database System*.