Personalized Location-based Brokering Using an Agent-Based Intermediary Architecture

Gaurav Tewari, Jim Youll, and Pattie Maes

MIT Media Laboratory
20 Ames Street, E15-305, Cambridge, MA 02142 USA
Tel: +1 617 253 7442, Fax: +1 617 253 6215, E-mail: {gtewari, jim, pattie}@media.mit.edu

Abstract

The Impulse research project at the MIT Media Lab, which captures our vision of the future of commerce, attempts to augment the rich sensory experience of the tangible world with the informational abundance, speed, and low search costs of the Internet. In this paper we present the design and implementation schema of a subset of the Impulse vision that addresses the problem of location-specific resource brokering. We discuss how two projects being conducted within the group, MARI (Multi-Attribute Resource Intermediary) and Wherehoo, can be combined to exploit the functionality afforded by a brokering architecture in a geographically constrained context. In particular, we examine the specific example of offering location-sensitive restaurant recommendations to mobile individuals.

Keywords:

Electronic Commerce; Location Sensitive Brokering; Software Agents; Intermediaries; Highly Mediated Communications; Product Brokering; Merchant Brokering; Negotiation; Utility Theory; Electronic Markets.

1 Introduction

In the physical world, a person enjoys a rich sensory experience, touches and feels physical objects, and participates in location-specific activities. Software agents, confined to the virtual digital world, enhance a user's "online" experience by proactively offering personalized assistance, and adapting to the user's needs over time, but generally do not also sense into the user's physical world. The Impulse [1] research project at the MIT Media Lab, which captures our vision of the future of commerce, combines the rich sensory experience of the tangible world with the information abundance, speed, and low search costs of the Internet. In this paper, we present a design and implementation schema of a subset of the Impulse vision. Specifically, we discuss how two projects being conducted within the group -- MARI [14], an infrastructure for the specification and brokering of heterogeneous products or services, and Wherehoo [13], a search engine optimized for location-specific searches -- can be combined to offer multi-attribute resource brokering in a geographically constrained context. In particular, we examine location-based decision support for recommending restaurants in the physical proximity of a mobile human user.

We are interested in facilitating *location-specific* resource brokering. *Physically proximal* buyers and sellers should be able to specify relative preferences and permissible tradeoffs along varying attributes of the transaction partner, as well as for the dimensions of the product in question. The motivation to tackle this class of problems arises from the fact that for many types of products and services, effective exchange and consumption requires that the buyer and the seller be located near one another. In particular, impulse shopping, such as for shoes or perfumes, and shopping for commodity items such as groceries or food, often necessitates that the actual product being purchased be available to the consumer almost immediately. In such a context, a person may be unwilling or unable to tolerate extended delivery latencies. The "need" for the product or service, if immediate, requires that the product be obtained within a constrained time span. In such a situation, it is desirable to exploit the functionality afforded by a brokering architecture with geographic constraints on the locations of the resources in question.

In this paper, we focus specifically on a system that matches consumers with restaurants located in their physical proximity. Our system can be viewed as an agglomeration of several entities: the mobile buyer, the buyer's agent, the MARI server (section 3.1), the Wherehoo server (section 3.2), seller agents, and physical sellers (see Figure 1). Using MARI's ability to extract utility functions and optimally match buyers and sellers, in conjunction with Wherehoo's ability to provide a geographically constrained listing of resource providers or sellers, we are able to optimally match a given consumer with a suitable product or merchant located in her physical proximity.

2 Related Work

This paper relates to work on location-based assistance in general, and specifically work within the domains of ubiquitous and context-aware computing [2], and use of learning agents with wireless devices [3]. Location-aware

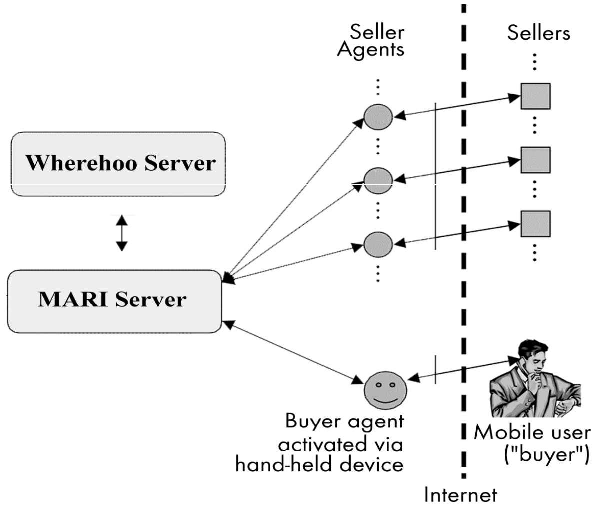


Figure 1: Major Components

applications are becoming increasingly popular as a consequence of the growing availability of consumer-oriented wireless networks and the emergence of locating technologies. Location information servers that attempt to simplify and offer infrastructural support towards the development of such applications by offering a set of generic location retrieval and notification services, are also being implemented with increasing frequency [15, 16]. This paper also relates to work in the area of shopping assistance and e-commerce decision support systems. Our work extends these previous expositions to introduce an intermediary infrastructure, capable of providing accurate brokering and decision support dependent on the user's physical location.

Our system relates to first generation price-comparison systems such as BargainFinder [6] and Jango [7], but goes much further than the rudimentary functionality afforded by such tools. We go beyond just bid and ask prices to include the attributes of the transaction parties as dimensions for consideration and differentiation. Our system relates to second generation value comparison shopping systems such as Personalogic [4], MySimon [20], and the Frictionless ValueShopper [5] in that it offers an advanced decision support engine, based upon multi-attribute utility theory, that meaningfully facilitates the exchange of complex and heterogeneous products. It differs from these systems in that it (i). allows both parties (buyers and sellers) to search for an optimal transaction partner, and (ii). it automates the match making between buyers and sellers.

Our system relates to online negotiation systems and auctions, such as Kasbah [9] and AuctionBot [12], and commercial systems provided by Moai [21], TradingDynamics [22] and others. It differs from them in proposing an integrative negotiation protocol and interaction model. This model, based upon bilateral argumentation, embodies an appropriate blend of formality and efficiency, and provides an alternative to the adversarial competitiveness of online auctions. Moreover, unlike most online shopping systems which generally operate in only one stage of the online shopping process [19], our system operates in three core stages -- namely product brokering, merchant brokering, and negotiation -- to provide a unified experience that better facilitates economically efficient and socially desirable transactions.

3 Research

3.1 MARI : Multi-Attribute Resource Intermediary

The MARI research project [14] in the Software Agents Group [8] at the MIT Media Lab proposes to radically improve online marketplaces that involve the buying and selling of goods and services. MARI is a generalized platform for the specification and brokering of heterogeneous goods and services. MARI makes it possible for both buyers and sellers to specify relative preferences for the transaction partner, as well as for the attributes of the product in question, making price just one of many factors influencing the decision to trade. MARI is unique because it allows both the buyer and the seller to exercise control. By allowing each party to associate weights with features from the underlying ontology, MARI makes it possible to take into account subtle differences in the characteristics of each party, to facilitate a more accurate match. MARI makes it possible for a buying agent to customize its bid price according to the characteristics of the seller, in addition to the dimensions of the product, and, conversely, for a selling agent to discriminate in its ask price depending upon the specific traits of a given bidder.

MARI addresses the "Product Brokering," "Merchant Brokering" and "Negotiation" stages of the standard Consumer Buying Behavior framework. In contrast to most commercial electronic marketplaces, MARI does not use an auction model as the negotiating framework. MARI tries to capture and quantify buyers' and sellers' intrinsic utility functions for a given product or service. This makes it substantially easier for participants in online marketplaces to accurately specify relative preferences and permissible tradeoffs within the context of a particular product domain.

MARI embodies a trend, expected to be key to the electronic marketplaces of tomorrow, in which negotiations become highly complex and participants engage in integrative negotiation over many aspects of a transaction. In this sense MARI appropriates features of the 'Market Maker' [9] and 'Tete-a-Tete'[10] projects, and extends them to create a more comprehensive solution.

MARI represents a general purpose architecture that is capable of supporting multiple sellers and buyers within multiple product domains. For the purposes of this project we specifically instantiated the MARI infrastructure in the domain of restaurant brokering. Hence, MARI was specifically encoded with a restaurant ontology and suitable complementary data.

3.2 Wherehoo

Wherehoo is a search engine [13] optimized for location-specific searches, with a web-based front-end for the addition of new records and maintenance of existing records. An infrastructure element of the Media Lab's Impulse project, it offers a socket interface for use by personal software agents that explore and interact with agents representing places in the physical world. As such, it is ideal for use by projects that use geographic location to personalize and filter information.

The primary function of Wherehoo is to act as a catalog that binds resources, points of interest, people or other features to geographic locations. One can query a given radius around some location – for example, a mobile user's current position (as is the case in our system). In response to the query, Wherehoo returns data about places within the desired search radius, with a vector to each place from the specified location. The records may be either pointers to more detailed information about the entries in the Wherehoo database (such as the URL of a restaurant's home page), or information that may be used directly by the querying system (such as an XML block that indicates "this is a US Postal Service mailbox").

3.3 Functional Overview and Application Domain

The present project involved creating an infrastructural "marketplace" that would enable users to participate in sophisticated location-sensitive resource brokering for products or services that require relatively quick or "just in time" delivery and consumption.

This "marketplace" is embodied within MARI. In this application, we use MARI in conjunction with the Wherehoo server, so that only sellers within some buyer-specified threshold distance from the buyer are considered as potential transaction partners. By using a highly mediated approach, we are able to successfully coordinate buyers and sellers in real-time to facilitate ad-hoc interactions.

A significant motivation for focusing specifically on a domain such as "restaurant brokering" is that it allows us to concentrate more upon the attributes relevant to the parties attempting to engage in the transaction without becoming overwhelmed by the material details of the product itself. In domains such as this, it is often the case that the characteristics of the seller actually define the product. In particular, we note that the food itself may comprise only one feature of the entire "dining experience," which may encompass such diverse dimensions as ambiance, conversation quality, nature of clientele, authenticity, quality of décor and service, etc. These dimensions do express the nature and quality of the "product" -- in this case the overall dining experience. Indeed, one can argue that in many domains in which the product or service being bought and sold is not as easily subjected to objective evaluation, the ability to be able to ontologically segregate and prioritize the various subtle impinging factors gains exaggerated significance and relevance. This argues for a "multi-tiered," not just a multi-attribute solution, in which we separate the attributes of the seller and the attributes of the product being sold into distinct tiers. Hence, the choice of restaurant marketplaces, in which "intangibles" form an integral component of the holistic product offering, is an appropriate and fitting choice as the target application domain for this project.

4 Implementation

4.1 Usage Scenario and Operational Schema

Figure 1 describes major components of the system. The "buyer agent" and "seller agents" are internal to MARI and may have been initialized by human users or by other agents. The "buyer agent" embodies the buyer's revealed preferences with respect to the desired resource. Similarly, "seller agents" embody the preferences and interests of sellers.

All interaction between buyer and seller agents is mediated by MARI, which evaluates the "cost" of each potential buyer-seller pairing. In our case, a 'market cycle' is triggered whenever a potential customer reveals her intent to find a suitable restaurant. At the beginning of the market cycle, MARI uses its internal mathematical approximation of the buyer's and sellers' utility functions to calculate "bids" and "asks." MARI attempts to optimize by selecting a pairing whose associated surplus is maximal among all possible pairings. The resultant pairing can be shown to be Pareto optimal [14, 18]. Effectively, the bid-ask *spread* indicates the surplus [17, 18] that the parties can derive from the transaction. As such, the buyer is "matched" with the seller with the lowest *ask*, among all sellers whose asks are less than or equal to the buyer's bid. The clearing price is set at the midpoint between the original bid and ask prices, equally dividing the surplus between the buyer and the seller. MARI thus acts as an impartial resource-brokering intermediary.

4.1.1 Capturing User Preferences

During the initialization stage, MARI gathers sufficient information from buyers and sellers so as to be able to effectively estimate their utility functions. This permits MARI to predict what valuation a given seller would associate with buyers who have not been explicitly valued, and vice versa. MARI's interaction with the user can be decomposed into several steps, enumerated below. Each ontology-specific attribute has a predefined "default" value associated with it, and the user can accept or override these defaults.

Step 1). Specifying the Ideal Offer (see Figure 2): The user specifies a "referential" or "preferred" configuration, or *offer*, which consists of specific product and transaction partner attribute values, as derived from the underlying domain ontology. The user must also associate a monetary valuation ("bid" or "ask") with this offer (referred to as the *pbsvalue*). Further, in order to exercise some constraint on automatically generated bids and asks, the user must specify the range (defined by a pair of highest and lowest endpoints) of permissible valuations (referred to as *maxvalue* and *minvalue*, respectively). This range corresponds to valuating the best and least qualified transaction partner, respectively.

In our implementation we actually allow the user to select from a set of generic, ontology-specific, pre-defined "profiles." Subsequently, the fields of the form in Figure 2 are automatically filled with reasonable default values, corresponding to the chosen profile. For example, predefined buyer profiles include those of "Media Lab Student" (*décor*, *food rating*, and *service* insensitive), "Media Lab Professor" (high *décor*, *food rating*, and *service* sensitivity), and "Media Lab Staff" (moderate *décor*, *food rating* and *service* sensitivity).

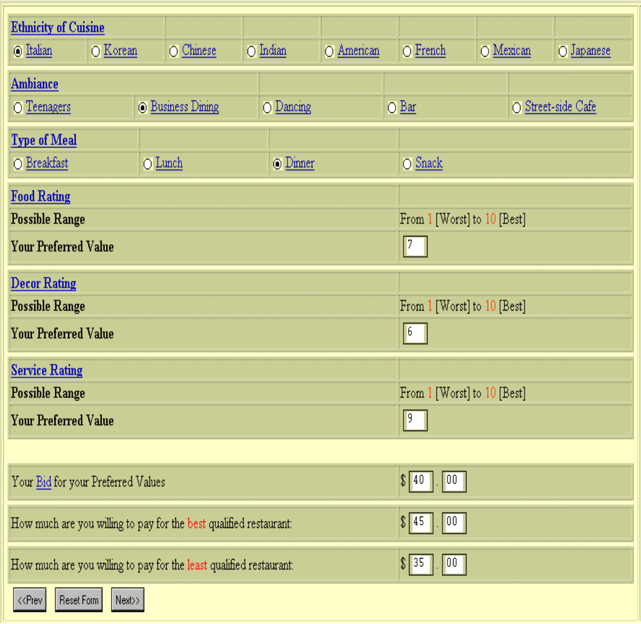


Figure 2: Specifying the "referential configuration" with an associated monetary valuation, as well as the range of permissible valuations

The attributes of any given product can be classified as being either fixed or flexible. A fixed attribute is one whose value, as specified by the user, is used for transaction party *qualification*. By contrast, flexible attributes have associated ranges, and are used for transaction party *valuation*.

For instance, in the context of a restaurant ontology (buyer's perspective[1]), we assume that relevant attributes include: *ethnicity* of the cuisine (Italian, Korean, Mexican etc.), *ambiance* (business dining, dancing, teenagers, street-side cafe), *type* of meal (breakfast, lunch, dinner or snack), *food rating*, *décor rating*, *service rating*, and *distance to restaurant* (in meters). We take *ethnicity*, *ambiance*, and *type* to be *fixed* attributes, meaning that the values of these attributes, as specified by the buyer, will be used for seller *qualification*. *Food rating*, *décor rating*, *service rating*, and *distance to restaurant,* by contrast, are *flexible* attributes, with predefined *ranges,* and are used for seller *valuation*.

[1] Conversely, from the seller's perspective, we define the ontological parameters that are relevant as: *number of people* currently present in the restaurant, level of *service* preferred by the buyer, and *type* of meal (breakfast, lunch, dinner, snack).

Each fixed attribute has a predefined set of *permissible* values, and the user must select an acceptable value from this set. For instance, the permissible values for *type* of meal might be "breakfast, lunch, dinner, snack."

Step 2). Gathering Ranges for Flexible Attributes (see Figure 3): Next, a user must also associate a permissible range of values with each flexible attribute. Again, depending on the profile chosen, these fields are filled in with default values, which the user can choose to override.

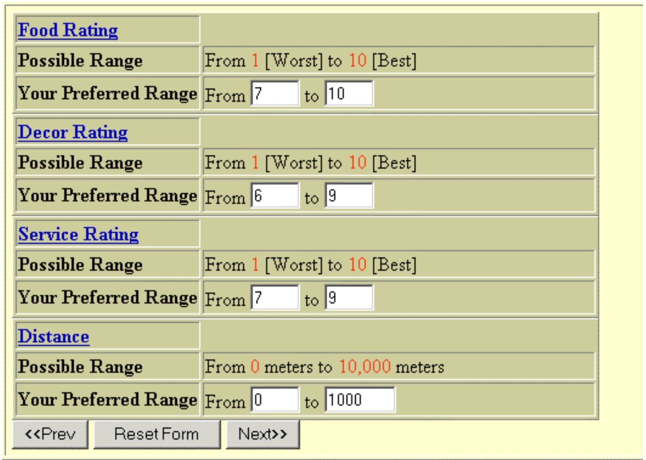


Figure 3: Specifying ranges for flexible attributes

Step 3). Inferring Attribute Weights: Having captured the above parameters, we automatically infer relative weights to be associated with each flexible attribute. The existence of weights is indicative of the fact that the user associates different degrees of importance with the various attributes, when evaluating a given offer.

Asking a user to explicitly specify weights for each attribute would no doubt be the most accurate and transparent technique. However, doing so imposes additional burden and tedium on the user. Moreover, it is not at all clear whether users themselves are able to accurately quantify these numerical relative preferences. We automatically derive relative "weights" for flexible attributes by using the heuristic that an attribute's weight or relative importance is proportional to how constrained its range of permissible values is, relative to the ranges of other flexible attributes. A tightly constrained range indicates that the user is relatively unwilling to compromise and hence the attribute is relatively more significant to her. With this in mind, we use the following formula to calculate the numerical weight factor to be associated with a given (flexible) attribute, p :

$$Weight(p) = \frac{1 - \left(\frac{\text{Permissible Range of p}}{\text{Possible Range of p}}\right)}{\sum_i \left[1 - \left(\frac{\text{Permissible Range of i}}{\text{Possible Range of i}}\right)\right]} \quad (i \text{ ranges over all flexible attributes})$$

Step 4). Quantifying Utility Functions: Corresponding to the user's chosen profile, "reasonable" generic utility functions are pre-associated with each flexible attribute (see

Figure 4). The generalized equation forms of these functions, in conjunction with the *pbsvalue*, *maxvalue*, and *minvalue* parameters specified by the user (in Figures 2 and 3), allows us to compute a mathematical approximation to the utility function corresponding to each flexible attribute.

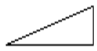
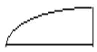
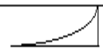
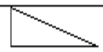
Attribute	Range	Utility Graph Shape	Utility Functional Generalization (a, b, c are arbitrary, non-negative constants)
Food Rating	7 - 10		(ax ± b)
Decor Rating	6 - 9		(-ax² ± bx ± c)
Service Rating	7 - 9		(ax² ± bx ± c)
Distance	0-1000		(-ax ± b)

Figure 4: Generic, profile-specific utility functions associated with flexible attributes

For example, let us assume that a given buyer is willing to accept a "service rating" ranging from 6 to 9. Assume that in her "referential offer" the buyer specifies a preferred value of 6. Further, given the buyer's chosen profile, say the market maker has pre-associated the function shown in Figure 4 with this flexible attribute as it varies over its range – the choice of this utility function would reflect the fact that the buyer is willing to bid higher as the service rating increases, and that her valuation increases exponentially as rating approaches the maximum possible. In this case we can derive the equation[2] which captures the change in the buyer's utility as reputation varies, as:

$$UF_2(x) = \left(\frac{maxvalue - pbsvalue}{(x_{hi} - x_{low})^2}\right)x^2 + \left(\frac{(-2)(maxvalue - pbsvalue)(x_{low})}{(x_{hi} - x_{low})^2}\right)x + \left(pbsvalue + \frac{(maxvalue - pbsvalue)(x_{low})^2}{(x_{hi} - x_{low})^2}\right)$$

Where:

x_{low} = the value of the attribute specified in the referential offer (i.e. 6);

x_{hi} = high endpoint of the permissible range (i.e. 9).

4.1.2 Delineating Transaction Partners

Our implementation relies on a version of MARI that operates in conjunction with Wherehoo. When a given buyer submits a restaurant demand request, MARI goes through two phases. In the first phase, MARI identifies the sellers who are *qualified* to meet the buyer's request. This corresponds to the subset of sellers who are able to satisfy the buyer's fixed attribute requirements. In the second phase, MARI uses its internal mathematical approximation of the buyer's and sellers' utility functions to calculate "bids" and "asks." For instance, given an arbitrary seller, s, we can computer buyer b's valuation or "bid" for s as:

$$\text{Valuation}_{b,s} = \sum_i f_i\left(x_{i,s}\right) * w_i$$

Where:

- i, ranges over all flexible attributes;
- f_i is the buyer's revealed utility function corresponding to attribute i;
- $x_{i,s}$ is the seller-specific value of attribute i;
- w_i is weight associated with attribute i by the buyer.

Then, for each qualified seller, MARI evaluates the "cost" that would be incurred if the buyer were to engage in a transaction with that seller. Currently, we take this "cost" to be equal to the "bid-ask spread," which can be interpreted as the aggregate surplus[3] [17, 18] that the two parties would derive if the transaction were to take place. We use this metric of "cost" since our indicator of the "goodness" of an allocation is welfare, which, in this case, is measured by the surplus that the allocation generates. Subsequently, the buyer is matched with the seller with whom the aggregate surplus of the two parties is jointly maximized

In general, we can conveniently formulate the problem of optimally pairing up the buyer with a seller as a "matching" problem. Mathematically, we can represent the state of the marketplace as a graph, G, in which sellers and the buyer represent distinct node sets. We refer to the set of buyers (singleton set) and sellers as B and S, respectively, and to the set of arcs as A. The buyer node $b \in B$ is connected to a subset of seller nodes (the *qualified* sellers) $S' \subseteq S$, via arcs of the form (b, s), each with associated arc "cost" c_{bs}. Given this formulation, our goal is to find a sub-graph $G' \subseteq G$, such that the sub-graph represents a feasible[4] pairing of the buyer with a seller with the largest overall "cost" (surplus), defined as the cost of its constituent arc. To accomplish this, our solution strategy mirrors that of a (modified) [5] *minimum* cost flow problem. With this formulation the matching problem can now be expressed as the following linear program [23]:

[2] This function is derived using standard algebraic techniques along with special properties of quadratic functions. In particular, we have used the fact that the global minima of a quadratic, of the form $y = ax^2 + bx + c$, occurs at $\frac{-b}{2a}$, that the quadratic function corresponding to the said utility function is monotonically increasing, and that the user has already revealed two data points, *(x, y),* on the curve: (x_{low}, *pbsvalue*) and (x_{hi}, *maxvalue*).

[3] Aggregate surplus is the sum of consumer and producer surplus. Consumer surplus is defined as the difference between the amount a consumer is willing to pay for a good and the amount she actually pays. Producer surplus is defined as the difference in the market price the producer receives for a good and the marginal cost incurred in its production [17].

[4] In this case the "feasibility" condition maintains that for a given buyer, the seller should be *qualified* to serve the buyer *and* that the buyer should meet the *qualification* criteria, if any, specified by the seller. Moreover, the buyer's bid can be no less than the seller's ask.

[5] Since we are actually trying to *maximize* the sum of our costs (aggregate surplus), we redefine costs in the min cost flow formulation to be the negative of the computed surpluses. Minimizing the sum of the *negatives* of the original quantities is equivalent to maximizing the sum of the original quantities.

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij} x_{ij}$$

subject to the constraints:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} = 1 \quad \text{for all } i \in S,$$

$$\sum_{\{j:(i,j)\in A\}} x_{ji} = 1 \quad \text{for all } i \in B,$$

$$x_{ij} \geq 0 \quad \text{for all } (i,j) \in A.$$

We solve this to identify buyer-seller pairings for which the aggregate surplus of transaction parties is globally maximized. The "clearing price" for any given transaction pair is, by default, set at the midpoint between the original bid and ask prices, thereby equally dividing the surplus between the buyer and the seller. The "market maker" can, however, modify this distribution of surplus, choosing to retain the bid-ask spread as operating profit for instance[6].

4.2 Technical Implementation

The client-side display device for our prototype is a Palm hand-held computing appliance equipped with a GPS receiver and a wireless Internet connection. The device is able to take user input and construct queries and commands that are understandable by the MARI server and, conversely, translate MARI replies and queries into human-readable form on the PDA.

Upon receiving a request from a buyer or buyer agent, MARI encapsulates the buyer's revealed preference profile and resource request. MARI also knows the physical location of the user, and uses this information to specifically attempt to find potential trading partners that lie within a user-specified radius from the user's current location. MARI accomplishes this by querying the Wherehoo server with the user's geographical location and a search radius. The Wherehoo server uses location as its primary criterion. In general, we will use the term "providers" to refer to physical resources such as businesses and services in the user's physical search domain, which also have an Internet presence and are registered with the Wherehoo server.

MARI issues queries to the Wherehoo server consisting of a location, search radius, and keywords. For instance, {<GPS location>, 1000m} is a query seeking all providers within 1000m of the <GPS location>. MARI then uses keywords to filter the list of physically proximal providers returned by Wherehoo in response to the original request. For example, the filtering keywords "Italian, Greek, food restaurant" are applied to seek restaurants catering Italian or Greek food. The MARI server then processes the data from the Wherehoo record to narrow the returned set to only those restaurants which are registered with MARI and, hence, whose preferences and attributes are well understood. For now, we accomplish this by requiring that restaurants put relevant MARI data in an XML block within a Web page, whose URL is embedded within the Wherehoo record corresponding to that restaurant.

Subsequently, this set of restaurants is further narrowed to those sellers who meet any other rigid qualification criteria specified by the buyer -- for instance, if the buyer is interested in having dinner ("*type* of meal" attribute), then at this stage MARI filters out all restaurants that do not serve dinner. Once the set of feasible potential sellers has thus been identified, MARI runs its matching algorithm to determine the optimal dining choice for the buyer. Finally, MARI notifies the user via the hand-held device that a match has been found (via an audible beep), and the buyer can then proceed to the restaurant. Symmetrically, MARI sends an alert to the seller (via an instant message or email). The details of the negotiation outcome are released to both the buyer and seller. The buyer knows how much he or she will have to pay, and what level of service and décor can be expected. Similarly, the seller knows what he or she is expected to provide and what compensation to expect. Furthermore, it is important to note that this entire process, from when the buyer submits her request to when she is notified of an appropriate match, occurs over the course of a few seconds.

5 Limitations, Problems Encountered, and Future Work

In general, we have faced a number of problems in being able to get client-side Palm application code to function adequately. We encountered problems in debugging our code on the Palm devices due to the propensity of buggy or problematic code to crash the devices. We had problems handling long strings, which made it difficult to parse long XML sequences that we used to pass data. To solve this, we created a small state machine within our application code to parse the XML. We are rewriting the application code to run on the KVM which will results in cleaner code overall. However, the KVM is still fairly slow compared to native code. It does not have floating point numbers, requiring code changes to simulate floating point using other data types. We experienced problems with networking that appear to be due to issues with the Java 2 Micro Edition (J2ME). In some cases, the network stack became corrupted or networking stopped unpredictably, requiring a hard reboot of the Palm device. In a few cases it was necessary to re-initialize the device before networking would function. Also, the J2ME applications do not behave like other Palm apps and do not save their state between sessions. Ordinarily, Palm applications can be resumed at the point at which they were interrupted, however code running under J2ME always seems to re-start from the 'beginning' when the application is called.

We hope to re-assess the implementation of the system and to have a fully functional demo completed in the near future. Also, it would have been hightly desirable to carry out an

[6] It is also worth noting that it is possible to further augment our matching process by incorporating more sophisticated negotiating techniques which could involve the back and forth exchange of proposals and counter-proposals between buyer and seller agents. This is an interesting topic, but one which we will defer for future research.

empirical validation of the applicability of the system and the proposed advantages. However, due to time constraints, this has been deferred as a topic for future research for now. Additionally, a big item on the agenda for future research is to carry out simulations. It would be fruitful to run simulations of cases of matches using carefully chosen parameters that would help us empirically substantiate the advantages of our matching algorithms as compared to other techniques and methods. Although, we have advocated the advantages of our approach at an intuitive, qualitative level, we would really like to focus our future work on gaining a better understanding of how much quantitatively better our approach is. Simulations would allows us to quantitatively compare alternative schemes, to understand under what circumstances things are optimized, as well as to identify the conditions that influence how and to what extent different parties experience tradeoffs, and how one party may benefit at the expense of the other.

Finally, we are also keen to carry out actual user studies that would allow us to gain an appreciation of whether our interface and approach is understandable and usable by actual buyers and sellers. Primarily, we intend to focus this work on gauging as to what extent users are able to convey their preferences and whether they find the process of doing so understandable.

6 Conclusion

In this paper, we have addressed the issue of *location-specific* resource brokering. This paper reflects our effort to formulate a system that amalgamates the efficiency and information abundance of the Internet with the tangible richness of physical shopping. We have discussed how two projects being conducted within our group, MARI (Multi-Attribute Resource Intermediary) and Wherehoo, have been combined to exploit the functionality afforded by a powerful brokering architecture in a geographically constrained context. Using the specific example of restaurant brokering, we have proposed the creation of an infrastructural "marketplace" that will enable users to participate in sophisticated location-sensitive resource brokering for products or services that require relatively quick or "just in time" delivery and consumption. The uniqueness of our contribution lies in the fact that we facilitate multi-tiered, multi-attribute brokering and valuation in a location-sensitive context, in which physical separation between the buyer and seller itself becomes a primary dimension for qualification and valuation.

7 References

[1] Impulse: http://agents.www.media.mit.edu/groups/agents/projects /impulse

[2] Billsus D., M. J. Pazzani, and J. Chen. 2000. "A Learning Agent for Wireless News Access." Proceedings of the *2000 International Conference on Intelligent User Interfaces*, p.33-36.

[3] Rhodes, B. J., N. Minar and J. Weaver. 1999. "Wearable Computing meets Ubiquitous Computing: Reaping the Best of Both Worlds." *Third International Symposium on Wearable Computers, IEEE Computing Society*: p.141-149.

[4] Personalogic: http://www.personalogic.com

[5] Frictionless: http://compare.frictionless.com

[6] Bargainfinder: http:// bf.cstar.ac.com/bf

[7] Jango: http://www.jango.com/

[8] Software Agents Group: http://agents.www.media.mit.edu/groups/agents/

[9] Wang, David Yi. June 1999. "Market Maker: An Agent-Mediated Marketplace Infrastructure." Master of Engineering Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA.

[10] Guttman, Robert H. September 1998. "Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce." Masters Thesis, MIT Media Laboratory, Cambridge, MA.

[11] Kasbah: http://ecommerce.media.mit.edu/Kasbah/

[12] Wurman, Peter R., Michael P. Wellman and William E. Walsh. May 1998. "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents." In *Proceedings of the Second International Conference on Autonomous Agents (Agents-98)*, Minneapolis, MN, USA.

[13] Wherehoo: http://wherehoo.media.mit.edu/about/

[14] MARI: http://www.media.mit.edu/~gtewari/MARI/

[15] Maass, Henning. August 1998. "Location-aware mobile applications based on directory services." *Mobile Network Applications* 3, 2, p.157 – 173.

[16] Jacob B. and T. Mudge. January 6-10, 1997. "Access to Local Resources in a Nomadic Environment." *USENIX Technical Conference*, Anaheim, CA.

[17] Pindyck, Robert S. and Daniel R. Rubinfeld. 1997. *Microeconomics*. Prentice-Hall, New Jersey.

[18] Varian, Hal R. 1999. *Intermediate Microeconomics*. W. W. Norton & Company, New York.

[19] Maes, P., Robert H. Guttman and Alexandros G. Moukas. March 1999. "Agents That Buy and Sell." Communications of the ACM, Vol. 42, No. 3.

[20] MySimon buyers guide: http://www.mysimon.com/consumer_resources/Buyers_ Guides/index.anml.

[21] Moai: http://www.moai.com.

[22] TradingDynamics: http://www.tradingdynamics.com.

[23] Ahuja, Ravindra K., Thomas L. Magnanti and James B Orlin. 1993. *Network Flows: Theory, Applications and Algorithms*. Prentice-Hall, New Jersey.