

The Architecture of a One-Stop Web-Window Shop

Mona Marathe[@]
and
Dr. Hemalatha Diwakar

Department of Computer Science, University of Pune
Ganeshkhind, Pune - 411007, INDIA

With the advent of e-commerce, web shops - where a potential buyer can make a purchase transaction on the web - are becoming commonplace. In everyday life, as buyers we face a typical need for matching or coordinating a host of articles/ services before taking the purchase decision. Very often the articles/services (items) which we seek to buy are marketed by completely different shops/ agencies, necessitating visits to different shops. The same situation translated to the web, means that the shopper has to make visits to several different web-sites and evaluate articles before taking the purchase decision. Just as in real life, as a first step the right web-shops have to be identified. Although, a search engine can do this, it would take a smart choice of search words to hit upon the "right" shops. "Right" in terms of the items they trade in and the geographical proximity to the shopper. Next comes the aspect of coordinating the chosen items. Clearly, this is very inconvenient because the various items cannot be compared simultaneously. To facilitate this requirement, what is needed is a Mediator, which provides a functional integration of the Web Shops of interest. Further, given the ubiquity of the requirement, a mediation mechanism, which is not only easy-to-use but also easy-to-create, easy-to-deploy and easy-to-maintain, is required. A novel yet simple mediator architecture is described here for mediating e-commerce sites founded on relational databases. The mechanism has been conceptualized to facilitate transactions in the B2C realm of E-Commerce.

Keywords : E-commerce, Mediator, Relational Databases, Web-Shopping, B2C, Query-by-Example, Prioritized Queries.

1. Introduction

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

Many shopping time scenarios require a co-ordination or matching of an assortment of services or goods before an exhaustive purchase decision is taken. The goods/ services (items) that are to be coordinated and purchased are very often sold by unrelated commercial enterprises. In fact very often the items themselves are semantically unrelated. The shopper understands the relation between them only in the context of their usage in day-to-day life. To clarify the point, consider the example of furnishing a house. Armed with this objective, a customer would first want to co-ordinate colour schemes, tiles, wallpaper, furniture, upholstery material, curtain fabric and artifacts before taking the purchase decision. Frequently, in the marketplace, business understandings exist among such establishments, made with the view of attracting customers as well as facilitating the purchase activity of the customer. For instance, the furniture store, upholstery store, curtain store may be housed in the same complex providing sale and discount schemes for a comprehensive shopper. However, on the web, each of the business establishments normally has independent e-commerce sites selling these goods/ services. Thus, currently in order to co-ordinate the various items, a customer has to first make a note of preferences, then visit the individual sites to check availability of the preferred articles. This activity is inherently iterative and the web-shopper may

[@] On study leave from CMC Limited of India

be forced to make several visits to the web-shop sites, before a satisfactory combination is isolated. There is also the possibility that the tedium involved in the activity may cause browsing fatigue and result in the web shopper making a less than optimal choice!

2. One-Stop Web-Window Shop

However, in this situation a mediated web site which stores integrated information about a chosen set of specific web sites - like furniture marts, paint shops, furnishing material stores - would be of immense use to a customer. In this case, one who desires to furnish his/ her home. What is needed is a kind of "**One-Stop WebWindow Shop**". Given the ubiquity of the problem, an integration mechanism is required which is not only easy-to-use but also easy-to-create, easy-to-deploy and easy-to-maintain.

A new methodology is proposed here, to provide a simple yet effective solution to this problem of integrating semantically unrelated sites that are related only in the context of their usage. This problem scenario is termed as "**Mediating Complementary E-commerce Sites**". The articles/services to be matched/ coordinated are termed as "**items**" and a suitable combination of items as "**combination**".

2.1. A Look at Shopper behaviour when shopping for Complementary Items.

Typically, when shopping for complementary items, a shopper embarks on this activity keeping in view certain priorities (must-have) and certain preferences (would-like-to-have). Initially, the shops selling the items are identified. Next, chosen items are compared to see if they complement each other. There is a typical shopper need of wanting to "see" all the choices at the same time (on the screen) in order to evaluate whether they complement one another. For example upholstery is compared with the colour of furniture before either of them is decided upon. Faced with the combinations available, the shopper may review and rework the earlier priorities/preferences resulting in new combinations. Furthermore, the shopper may decide upon a subset of the items from the wish-list, while eliminating items not fitting the bill and augment the purchase with other complementary items.

2.2. What the Web lacks to support such behaviour

At present, the Search Engine is the most ubiquitous tool available to Web-Surfers. However, even the best Search Engine only helps identify the e-commerce sites selling items of the complementary set. Each of the sites would then have to be visited separately by the shopper. If the shopper desires to retain information about the items from web sites visited (for the purpose of comparison later) then the web pages would have to be saved to the local hard disk. Moreover, such an action may not help given that it is difficult to save dynamically generated web-sites and that saving a web-site to disk does not necessarily save its pictures or anything except its html text. An effort at comparison of the various items would entail bringing up these stored pages in separate windows and tiling them together on the screen for viewing. This again may require considerable effort on the part of the shopper for tiling and re-tiling multiple windows coherently within the screen area.

Another aspect of the problem is the necessary locality of the solution desired, juxtaposed against the global nature of the substrate - the Internet. A majority of shopping, particularly that undertaken by individuals, is local in nature. Which is to say that one of the attractions associated with a commodity would be - the proximity of the shop selling it and the currency in which it is sold. Individuals would mostly avoid paying in foreign currency. Although credit cards sometimes take the sting out of paying in foreign currency, the higher shipping costs and the higher risk of non-delivery - not to mention the complex nation-specific laws governing import -continue to detract the web-shopper from attempting to shop at geographically distant web-shops. Describing these constraints to a search engine in a few search terms is a difficult if not impossible task.

2.3. The Proposed Mediator for Complementary Web-Shops

The **One-Stop-WebWindow-Shop** has been conceptualized with a view to eliminate the above shortcomings. In reality, the WebWindow-Shop is a web-site accessible by means of a URL. The solution comprises a 3-tier architecture, integrating the complementary e-commerce sites (Tier-1) with the web-shoppers (Tier-3) by means of a mediator - the **One-Stop-WebWindow-Shop** (Tier-2).

The complementary e-commerce sites, referred to as Web Shops henceforth, are assumed to be founded on web-enabled relational databases (which is the prevalent scenario) and accessible by means of JDBC/ODBC connectivity. The mediator itself comprises a database and a query application hosted at a web-server and accessible by means of a URL.

The mediated database contains the integrated schema, which is formed by importing the schemas of the identified web-shops.

The applet-based query application functions atop the mediated database and provides the web-shopper with a seamless view of the web shops of interest. To facilitate the shopper behaviour described

earlier, a novel method of **Prioritized Unified Querying** and **Summary Querying** is implemented in the User Interface part of the query application. This method facilitates the shopper in providing priorities and mixing and matching successive results in order to isolate the most suitable combination items.

A competent team of authors of the Mediator is central to the conceptualization of the One-Stop-WebWindow-Shop and is referred as the **Mediator Author**. The Mediator Author is responsible for

- identifying the clientele,
- identifying the web-shops of interest to the clientele. Identifying the web shops would not only involve evaluating the complementary nature of the items sold but also other basic real-life issues such as the reliability of the web-shops and the ability of the web-shop to realize sale in the geographical area of the shopper (dispatch, payments issues).

The nature of this role - requiring essentially human acumen - forecloses the possibility of the team being composed of software agents even at the current state-of-the-art.

2.4. Focus of the One-Stop-WebWindow-Shop

The focus maintained while conceptualizing this mechanism is

- to deliver a simple yet effective mechanism for integrating different e-commerce sites founded on web-enabled relational databases in order to facilitate the process of collaboration between complementary web-shops who so desire it.
- to provide an effective and efficient query application for the use of the web-shopper

2.5. Assumptions Made

Keeping in view the focus, certain assumptions have been made regarding the features of the relational databases. These are

- **The databases are completely homogeneous – schematically and content-wise.**

The following assumptions are made:-

1. All the web resources will be relational databases with JDBC connectivity.
2. The source database will provide complete access.
3. Schemas representing similar real world entities will be identical. For example, all furniture marts will have same schemas.
4. Schemas participating in the mediated web site will agree completely in the data content. (In spite of identical schemas, the data representations of items within the schematic structure can differ enormously. Typically, differences occur due to - difference in granularity of data content, difference in scale assumed, difference in naming. However, for the purpose of this effort it has been assumed otherwise)

[1], [2], [3], [4] are some of the research efforts aimed at resolving these issues. [5] shows that it is not just marketers, but also customers that are reaching out to solve this problem by constructing electronic malls in order to work with the many differences in the way vendors categorize their product XML and database schemata.

- **The schemas of the source databases are unchanging**

Given the changeful nature of the web, it is inevitable that web resources will undergo changes. Changes may be in the form of elimination of a web resource, changing of the schema and addition/deletion/modification to the content. Of these the changes of the first two types will make a direct impact on the suggested methodology. However, here it has been assumed that the schemas will be unchanging. However, it is appreciated that while working on the web this situation is inevitable and the next phase of this research effort will be aimed at tackling this problem.

Restrictive though these assumptions may seem, they form a very sound base on which to build the kernel of the mediator. However, for this Mediator to be viable, sources other than strictly defined relational databases will have to be brought into its fold. This can be achieved by building an outer shell around the mediator kernel based on the emergent XML standards for e-commerce (such as RosettaNet [6] and Common Business Library [7]) and wrapper-parser techniques. Once a standard schema is identified for the mediated database, the onus of defining covering XML pages describing the interpretation from their own proprietary content to the Mediator standard can be borne by the source sites. Browsers supporting XML parsers are slowly but surely becoming the norm. In this scenario, the assumptions made can be overcome with some effort. The effort of integrating heterogeneous databases under one umbrella schema has been getting attention in the area of Federated databases ([8] is one such research initiative). Representing relational schemas in XML has been covered in [9].

3. The Architecture of the One-Stop-WebWindow-Shop

Figure-1 depicts the architectural overview of the One-Stop-WebWindow-Shop. The functionality is described further.

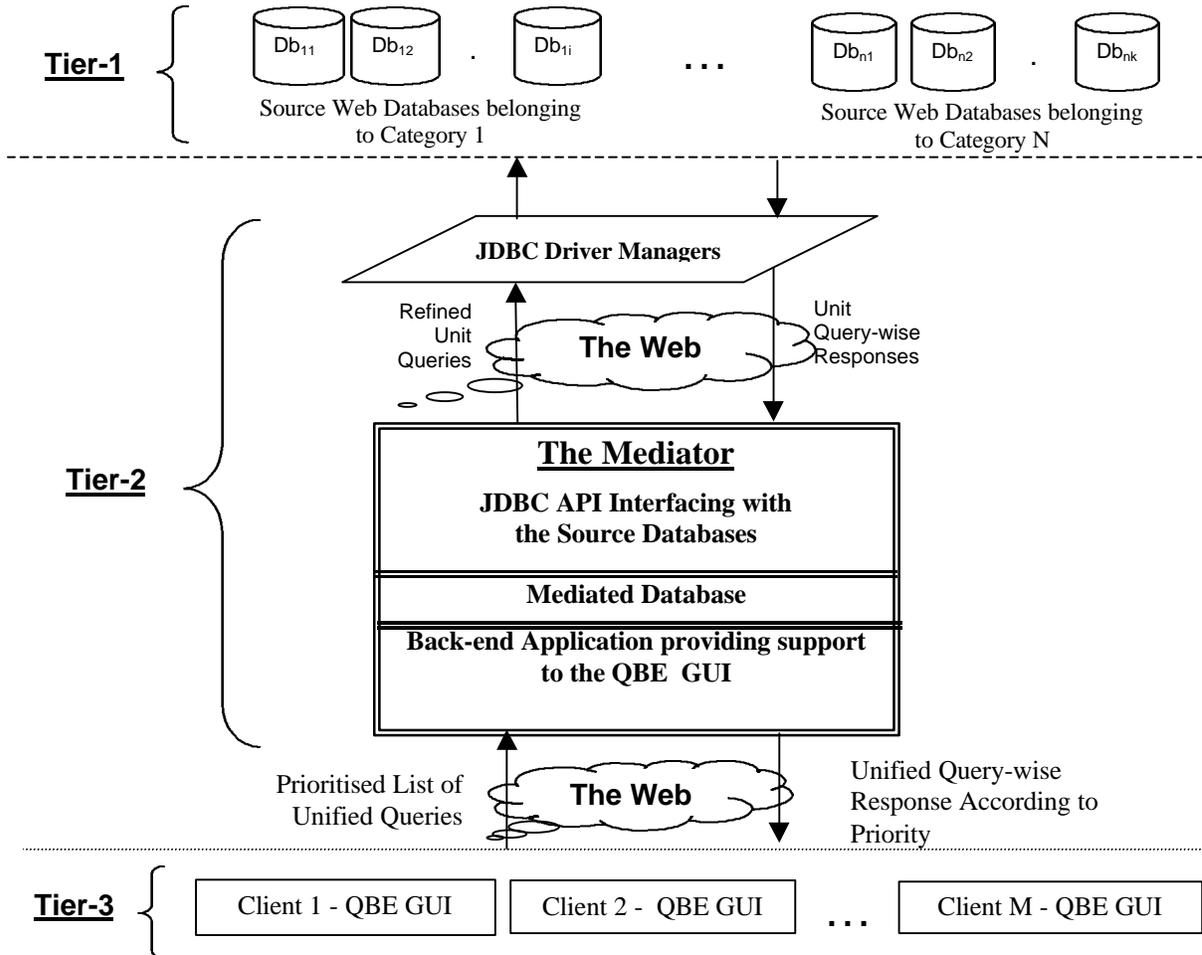


Figure-1 : The Mediator Architectural Diagram.

3.1. The Mediated Database

The mediated database has two parts. The first is the **Common Database** named as such because the data stored here would be common to all the clients of the mediator. (The word "Client" has been used to represent web-shoppers actively accessing the mediator through a browser.) It would store the integrated schemas of the source databases, meta data about the source databases, the data dictionary, extracted master data and query filters. The second part, the **Temporary Database (workspaces)** is maintained client-wise and would store the list of queries fired by a client and the responses extracted from the sources for the same.

As clarified in the assumption about Homogeneity of Schemas - several source web databases belonging to the same category will share an identical schema. We refer to such a schema as a **Database Category**. Thus, there may be a number of source web databases mapping onto a single database category. A DB_ID - a unique identifier - is generated for each source database. It is likely that different items from distinct source databases may share codes, which would result in overwriting of records in successive extractions. To overcome this, the schemas of participating databases are uniformly extended with the DB_ID .

The Mediator Author may possess specialized information regarding some of the source databases. These are stored as specialized facts in the Common Database and are referred to as **Query Filters**. For instance, it may happen that a particular Furniture Mart sells only Teak furniture. This information is recorded as specialized facts and is used by the query application to eliminate destinations,

which will yield a *null* result for a specified query. For instance, if a particular web-shopper asks for cane furniture, the query should not be directed to the mart selling only Teak furniture and therefore should be eliminated altogether from the list of destination source database to which the query is dispatched, when this can be done. This helps in increasing the speed of query execution.

3.2. An Illustrative Example

This concept is illustrated with an example - a Mediator for Furnishing called "**The One Stop WebWindow Shop for Home-Furnishing**". In this example the mediator integrates the e-commerce sites for

- Furniture Marts
- Furnishing Material Stores

Thus, the database will contain only 2 different categories of database schemas. However, the hypothetical source sites integrated are as follows:

- 3 Furniture Marts : "Only Teak", "Home Furniture", "Creative Cane";
- 2 Furnishing Material Stores : "Ethnic Cottons", "World of Upholstery"

Here the number of integrated databases will be $(3 + 2) = 5$ databases.

Let us assume that the two database categories have the following schemas:

Schema Type : Furniture Mart

Furniture Type Master (Item_type, Description, Specifications) : Storing information about generic furniture items like Lounge-sets, dining sets etc,

Wood Type Master (Wood_type, Description) : Storing information about the type of wood in which furniture is available like - teak, rubber wood etc

Item Master (Item_code, Item_type, Wood_type, Cost, Picture) : Storing information about physical furniture items available along with the picture.

Schema Type : Furnishing Material Store

Furnishing Material Master (Material_type, Description) : Storing information about the material such as Cotton, Poplin, Lace etc

Article Type Master (Article_type, Description) : Storing information about the types of articles for which the material could be used. For eg : Curtain Cloth, Upholstery Material, Wallpaper etc

Print Master (Print_type, Description) : Storing information about the variety of print - such as Floral, Geometric, Abstract etc

Item Master (Item_code, Material_type, Article_type, Print_type, Colour, Picture, Cost_Per_Metre) : Storing information about the items available and their cost.

For the described schemas, the **Query Filters** would be:

- "Only Teak" : Furniture_type_master.wood_type = "TEAK"
- "Creative Cane" : Furniture_type_master.wood_type = "CANE"
- "Ethnic Cottons": Furnishing_material_master.material_type = "COTTON"

3.3. The Query Application

The application is the second part of the mediator architecture. It is built atop the integrated database and hosted at the same web site. The focus of the application is to provide a means by which a web shopper can seamlessly query a group of web shops and simultaneously view the information retrieved. The application comprises a browser based front-end applet, which implements a Query-By-Example Graphical User Interface. [10] gives a description of QBE functionality. The applet is supported by a server-based application that implements the functionality required for processing the query, input at the applet and generating the required response.

3.3.1. The Front-End: The Query-By-Example Graphical User Interface (QBE GUI)

The front-end supports the following main features to meet the typical shopper needs mentioned earlier.

Feature No. 1 : Prioritized List of Unified Queries (PLUQ).

As a first step, windows are provided wherein queries can be formulated on each database category individually. These are called **Unit queries**. A Query-by-Example mechanism is provided here. Queries that specify join conditions on tables from distinct database categories are **not** allowed at this stage. This constraint has been imposed because identifying database category-wise unit queries from the specified join query - mandatory for query processing - would entail decomposition of the query - a complex research problem in itself. A single priority number can be associated with queries from distinct database categories, thus uniting them into one cluster. These are termed as **Unified Queries**. Several queries may be specified on the same database category for the same priority number. These are considered to be "OR" queries and are termed as **Vertical Queries**. Several Unified Queries may be specified each

having a different priority number – giving rise to a **Prioritized List of Unified Queries** (PLUQ). This list is submitted to the back-end server application. To clarify these concepts, a sample Query Session on the “One-Stop WebWindow Shop for Furnishing Material” is delineated here:

Consider a shopper shopping for home furniture and matching upholstery. As a first priority the shopper is interested in Teak Wood Dining Sets and Lounge Sets and optionally Rubber Wood Dining Sets and Bedroom Sets. In either case the shopper would like to have satin upholstery with floral design.

As a second priority, the shopper is interested in viewing Cane Furniture - Dining Sets, Lounge Sets and Bedroom Sets but this time the upholstery desired is Cotton with a Geometric Pattern.

The Table-1 represents the Query-Session-1.

Table-1 : Query Session - 1

<i>Priority No.</i>	<i>Furniture Mart</i>	<i>Upholstery Mart</i>
<i>1</i>	(Wood Type = “Teak”) AND (Set Type in (“Dining Sets”, “Lounge Sets”))	Satin with Floral Design
	(Wood_Type = "Rubber Wood") AND (Set_type in ("Dining Sets", "Bedroom Sets"))	
<i>Results Retrieved into Temporary Tables:</i>	First into S1_P1_C11 Second into S1_P1_C12	S1_P1_C2
<i>2</i>	(Wood Type = “Cane”) AND (Item Type in (“Dining Sets”, “Lounge Sets”, “Bedroom Sets”))	Cotton with a Geometric Pattern
<i>Results Retrieved into Temporary Tables:</i>	S1_P2_C1	S1_P2_C2

The naming convention of the temporary tables into which the responses are retrieved and stored in the temporary database is $S_i_P_j_C_k[l]$. Where:

- i denotes the session number for the client
- j denotes the priority number. of the query
- k denotes the database category no.
- l denotes the serial number within the list of vertical queries (if any).

In this example, S1_P1_C11, S1_P1_C12 represent **vertical queries**. These queries are formulated as independent queries (and not clubbed into a single SELECT with an OR clause) because the list of attributes retrieved in each case may be different based on the client's choice. Thus, they are retrieved by means of different queries and stored into separate temporary tables. To clarify the fine difference, consider these two vertical queries formulated on the Furniture Mart:

Vertical Query 1: SELECT item_code, item_type, wood_type, cost, picture
FROM Item_master
WHERE Wood_type = "TEAK"
AND Item_type in ("DINING_SET", "LOUNGE_SET")

Vertical Query 2: SELECT item_code, item_type, wood_type, picture
FROM Item_master
WHERE Wood_type = "RUBBER_WOOD"
AND Item_type in ("DINING_SET", "BEDROOM_SET")

In the first query the client retrieves the attribute "cost", while in the second query "cost" is not chosen. All the 5 query elements (including the two vertical queries) described in the table above are unit queries. Together they form the Prioritized List of Unified Queries.

Feature No. 2 : Database Category-wise Result Window and Priority-wise Tiling

Each of the results retrieved is displayed in the order of priority. Each database category is retrieved into a separate window so that all the records retrieved for a particular priority can be seen together by simply tiling all the windows of that priority on the screen.

Feature No. 3 : Summary Queries

Each of the queries specified by the user result in the creation of a temporary table, which stores the response to the query. This table is stored in the temporary database at the mediator. The client can specify any kind of query on these tables without adhering to the earlier restriction debarring joins on fields belonging to tables of different database categories. These queries are again operated priority-wise, but the

data on which this query is operated is limited to the data retrieved in the earlier session. To clarify, consider the following query session:

Query Session - 2:

The shopper finds that both the furniture options specified in priority-1 fit the requirement. Now, the question is which one has a matching upholstery colour. Here the client desires Teak Furniture from the ones selected earlier matched with Cream-coloured Satin Upholstery OR Rubber-wood furniture with rust-coloured Satin Upholstery. At this stage since both the furniture and upholstery are fetched into the temporary tables at the mediated database a single query can specify joins between fields of tables belonging to different database categories. Therefore, the query is formulated on the temporary tables: S1_P1_C11, S1_P1_C12 and S1_P1_C2.

The application works on the temporary database which stores the results of all the earlier queries specified by the client in the form of temporary tables, to yield the desired results. In this case, a subset of the records retrieved earlier will be displayed satisfying the stated condition.

3.3.2. The Back-End Engine: Server application

The PLUQ (Section 3.3.1 Feature 1) or the Summary Query (Section 3.3.1 Feature 2) – as the case may be - is directed to the **Mediator Server Application** residing at the web-server hosting the Integrated Schema. The Server Application is composed of several processes. The application internally retains a copy of the PLUQ and stores it into the temporary database. It first checks if the query is to be executed locally (as explained in the example given in Query Session-2). Subsequently, the process separates the Unified Query into individual database category-wise queries. These category specific queries termed as **Unit Queries** are checked destination-wise, in order to filter out redundant destinations. This yields a list of destinations - Unit Query-wise. Based on this list, the Unit Queries are dispatched to the appropriate source databases using the JDBC API. The responses received are stored into the new temporary table created in the temporary database. On successful completion of the activity of executing Unit Queries, the underlying SQL Engine is called for executing the Unified Queries on the now populated temporary database. The calls are made based on the user-specified priority. The responses delivered by the SQL engine are finally delivered to the front-end QBE GUI of the client.

4. Conclusion

The problem of providing a "One Stop Window Shop" for Complementary E-commerce sites necessitates a simple, cost-effective mediation mechanism which is not only easy-to-use but also easy-to-create, easy-to-deploy and easy-to-maintain. We have proposed a methodology of integrating schemas of the various web resources at a mediated web site and providing the web-shopper with a QBE GUI to facilitate querying. Although the methodology suggested does not lead to a complete solution, it is inherently both scalable and extensible. A prototype of this system is being implemented at present, using MS-Access for the creation of the back-end Mediated Database and Java for the programming of the applet and the back-end functionality for communicating with source databases. We plan to extend the usability of the mechanism by

- Enhancing the scope by bringing transactions such as purchasing/ ordering/ booking of items across the Complementary E-commerce Sites within the scope of this architecture.
- Allowing for schematic changes in source databases. As described earlier, this can be achieved by the design of a standard for defining the covering XML page and a standard for the mediated schema. Once this is done, the onus of describing the mapping from the proprietary content to the standard schema can be borne by the source sites. In order to test this concept, we have succeeded in extracting the schema and content of a remote relational database based on a covering XML page that stored representations of the same. An XML parser was used to extract the schema and then a database was created. An Object-Relational Converter developed by us in Java was used to extract the database content from the XML page, which was then stored in this database.
- Automating the process of implementation of the mediators. One of the aims of this effort is to suggest a methodology that was easy-to-create. In fact quite a few of the processes undertaken in the creation of the mediator are such that they have to be repeated at every instantiation. These processes include - extraction of schemas, extraction of data, augmentation of schemas (with DB_ID), generation of metadata and creation of the QBE GUI application. We intend to create tools for the automatic generation of each of the processes above so that a minimum amount of input will yield the desired results.

Such a mediator architecture will facilitate the implementation of a revenue model in the form of another shell around the mediator kernel. By virtue of being located between the shopper and the web shop,

it would be possible to keep a track of the transactions performed on individual web shops. This statistic could form the basis of an appropriate revenue model, which would render the Mediator commercially viable. Also, marketing schemes such as slab-based discounts given on package purchases across web-shops could be implemented and the benefit passed on to the shopper. In its evolved form, such a mediator architecture would emerge as a demand/need-specific marketplace which will enhance value for the shopper as well as the web-shop.

Commercial viability also brings into focus the wisdom in using front-line, high-end RDBMSs such as Oracle 8i or Informix Web Solution. The use of such established packages would facilitate the creation and deployment of several One-Stop WebWindow Shops for home furnishing, house construction, travel planning, grocery shopping etc, at one host, thus enhancing the commercial viability of such an undertaking.

5. References

- [1] William W. Cohen, "The WHIRL approach to integration: An overview.", Proceedings of the AAAI-98 Workshop on AI and Information Integration. AAAI Press, 1998.
- [2] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman et al, "The TSIMMIS Approach to Mediation: Data Models and Languages", Journal of Intelligent Information System, 1997
- [3] C.A. Knoblock, S. Minton, J.L. Ambite, et al, "Modeling Web Sources for Information Integration", Proceedings of the 11th National Conference on Artificial Intelligence, AAAI Press, 1998
- [4] "MIX : Mediation of Information using XML", <http://www.npaci.edu/DICE/MIX>
- [5] EMALL, <http://www.state.ma.us/emall/>
- [6] RosettaNet, <http://www.oasis-open.org/cover/rosettaNet.html>
- [7] Common Business Library (CBL), <http://www.oasis-open.org/cover/cbl.html>
- [8] J. Hammer, "Resolving Semantic Heterogeneity in a Federation of Autonomous Heterogenous Database Systems", <http://dbpubs.stanford.edu:8090/pub/1994-11>
- [9] C. Baru, "Xviews: XML Views of Relational Schemas", Proceedings of the 10th International Workshop on Database & Expert Systems Applications, *The Computer Journal*, 40(5), 1997
- [10] R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems, Third Edition.", Addison Wesley, 2nd Indian Reprint 2000, Chapter 9 : ER and ERR-to-Relational Mapping, and Other Relational Languages, Section 9.5 : Overview of the QBE Language, Pgs310-317