User Modelling for Information Retrieval from Multidatabases

Lachlan Mackinnon
Herriot-Watt University
Riccarton
Edinburgh EH14 4AS, UK
+44 131 451 3410
lachlan@cee.hw.ac.uk

Michael Wilson
. Rutherford Appleton Laboratory
Chilton, Didcot
Oxon OX11 0QX, UK
+44 1235 44 6619
M.Wilson@rl.ac.uk

ABSTRACT
The design options in resolving heterogeneous data source access from a single query, and for a supporting user modelling component are discussed. The MIPS system is used as an example to consider the role of user modelling in automatically generating hypermedia presentations of the information retrieved from such distributed data sources whose semantics are unknown to its users. A user modelling component from a previous system was able to be used, showing its portability. The opportunities for the application of user modelling to tailor the retrieval and presentation process are investigated, not only for query construction and information filtering, but throughout the entire process.

Keywords
user modelling, multidatabase, intelligent interface

INTRODUCTION
The traditional use for user modelling in information retrieval [1] and filtering [2] tasks had been to establish a description of users' interests to supplement queries. User models are generally acquired actively by asking users to select terms of interest to them, and passively by collecting the terms used in past queries [8].

Recently several information retrieval systems [3, 12] have included ontologies of domain terms which include richer sets of links between them than merely synonymy, and which therefore support richer inferences than merely adding synonyms to the set of search terms. One such modern information retrieval system is MIPS [4,5,6] which will be used in this paper as the exemplar.

The motivation for the development of these systems is that they allow users to query for information by content rather than location. For example, querying by location is done in the database language SQL in which information requests are expressed in terms of the columns required from tables within named databases; equally the World Wide Web (WWW) uses Universal Resource Locators (URLs) to identify each object by its location. As multiple databases are available over networks, there is a need for multidatabase solutions which allow single queries expressed in terms of content semantics to be mapped to the pertinent set of columns in tables within named databases, and as the WWW grows the need for index tools which map semantic content terms to sets of URL locations is being met by various web robot based index sites (e.g. AltaVista, Lycos etc…).

The structure of these information retrieval interfaces is to support the eight stages of :

1) to expand the query provided in order to enrich the information description;

2) to map this description to the largest possible set of stored descriptions of available information sources;

3) to reduce the number information sources according to processing constraints of redundancy, time, cost and reliability;

4) to retrieve the currently available information;

5) to constrain the amount of returned data if there is too much;

6) to convert the returned data into common formats;

7) to resolve any conflicts within the data set;

8) finally to design the presentation of the retrieved information.

At nearly all of these stages it is possible to merely return to the user and ask them to explicitly perform the required actions themselves (with varying support from tools). Systems such as MIPS are designed to act as intelligent user interfaces to the set of information sources, and perform as much as is practical of the process themselves, only returning for dialogue with the user when they cannot reach a conclusion.

This paper describes the role the user modelling component of such an intelligent user interface can fulfill in supporting this information retrieval process across multiple information sources. Firstly the design space of possible user modelling components is outlined. Then next section outlines the problem of retrieving information from multiple information sources and the multi-process architecture of the MIPS system designed to address it. Then the algorithm that incorporates inferencing across a domain ontology in order to retrieve information is described in more detail before stepping through the eight stages of the information retrieval process and describing the support that user modelling can offer each. It is intended that this account should guide others engineering intelligent user interfaces for the retrieval of information from multiple information sources as to the design options available for tailoring both retrieval and presentation to individual users.

THE USER MODELLING COMPONENT DESIGN SPACE

The general inference supported by user models is that of heuristic classification [7]. Heuristics allow individual users to be classified, and then domain properties inferred about individual users on the basis of that classification. To support these inferences, rules must exist which associate attribute values with user classes, in order to determine to which classes users belong. As new information is established about users, these rules will be triggered, and the user's classification will both grow more rich, and change from one class to another. A second set of rules must exist to relate user classes to domain classes. A general inheritance rule is usually applied for domain properties. When a user model is impoverished domain property consequences would also be impoverished unless defaults were applied which could later be overridden.

The rate at which new user models become populated depends on the methods used for acquiring information about the user. Active acquisition of information can be achieved through users completing detailed questions prior to using an application. However, this places a stage in the use of an application which does not appear to users to be directly required to meet their goals, with the result that they either skip over it, or regard the application as less usable. Consequently, most applications use minimal direct questions balanced to acquire essential information about the user (e.g. user language) supplemented by passive user knowledge acquisition components which establish increasingly rich user models as use progresses by monitoring the users interaction with the application [8].

User modelling systems usually include a method for classifying users into groups, and relating these groups to domain concepts, and a software user modelling component to be incorporated into an application. User modelling components generally include a component to store the user model, and an interface to write and read from that store. The write interface supports the acquisition of user models, and the read interface supports the querying of the user model for domain or application information. The store itself supports the heuristic classification either at write time or read time. If the inferencing is performed at write time then it will be performed over all information stored, but such inferences can normally by performed in a parallel thread to the main process and although they add massively to overall processing, they do not delay user response time. However, conclusions drawn from inferences made at write time may become inconsistent with statements written later, or their consequences, therefore a truth maintenance or non-monatomic reasoning component is required to resolve these inconsistencies. If inferences are drawn at read time, only those inferences required to support the read operations are drawn, and the results of the inferences do not become inconsistent with other knowledge about the user. However, inferences at read time will cause the main process to pause since the read function returns, thereby delaying system response time to the user.

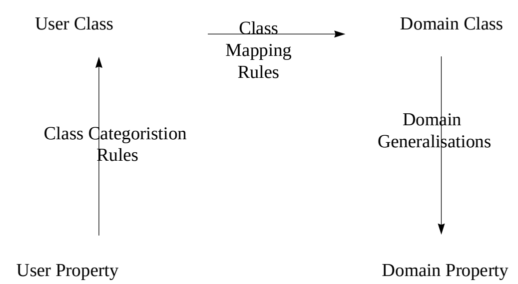

Figure 1 - The heuristic classification inference performed by user modelling components

Complex user modelling components have been developed which can be incorporated into natural language dialogue systems [8]. Simpler user modelling components allow descriptions of users and their preferences to be registered and used to tailor user interface characteristics for them [9].

THE MIPS SYSTEM

The MIPS system supports a single user query which can retrieve information from multiple information sources and yet provides a single integrated answer to the user. The information sources are assumed to be heterogeneous in that they include text retrieval systems, WWW sites, and relational. There can be no assumption that the databases have been developed as distributed databases using common schema or units, but may use idiosyncratic terms to identify data. Further, the information to be retrieved will include structured numeric data and strings, but also text, image, sound and video files requiring multimedia presentation technology. It is assumed that some of the information sources are freely available in the user's site, others are freely available elsewhere, while still more are available at some access cost at various locations around the world. To add to the breadth of the requirements, the users could vary from both domain and computer naive, to both domain and application experts, with the necessity for the user interface to adapt to their varying needs and abilities.

The solution adopted was to develop a multi-process system where a general query tool provided a user interface for composing queries based on a presentation of the available terminology, and a syntax checking dialogue interface. Completed queries are passed to a selection and retrieval component which negotiates with a knowledge based system (KBS) to produce a set of queries aimed at each target information source that are then dispatched by a communications process. The replies are received back and passed up to the selection and retrieval module which again interacts with the KBS to resolve data conflicts and integrate the set of replies. The integrated replies are then passed to a web builder which constructs a hypermedia web from them, that is then passed through a presentation manager (that has the ability to store it) to presentation tools. The hypermedia web presentation tools allow the user to browse through the web in order to view all aspects of the returned data. The overall architecture of the system is shown in figure 2. The implemented system was demonstrated using external information sources and internal metadata for the domain of tourism in Greece.

The KBS in MIPS contains 4 components. Firstly, interfaces to the processes which call upon it to perform functions; secondly, a domain modelling component; thirdly a context modelling component that stores details of the current session, and fourthly a user modelling component. The interfaces to other processes that perform functions, call upon the other three components to support those functions. The user modelling component and the interfaces involved in the selection of information will be described below. Although the domain modelling component is described elsewhere [5] it is necessary to outline its structure here as a basis for a description of the retrieval process.

The domain model represents generic concepts using a simple semantic network including links for object subtyping and instantiation, component parts of, geographic location of, attribute possession and value. The ontology represents about 400 high level generic concepts on the assumption that this can either be extended for any particular application domain or linked to a specialised ontology server for a domain. The conceptual domain model does not attempt to capture the whole or even a substantial part of common sense knowledge, and if an ontology becomes available which serves that purpose then the current one could be replaced by it [10]. Similarly the domain model has not been optimised for either ontological generality or performance efficiency, although it has been used to explore tradeoffs in these qualities as well as the required maintenance effort. A distinction is drawn in the ontology between conceptual types (e.g. city, town, airport) and ground data types (e.g. string, integer, video). The semantics of conceptual types are defined by their links with other types, over which inference can be drawn. Ground data types denote entries in information sources. Types and instances inherit the attributes of parents through the ISA link at run time. For example Figure 3 shows a pseudo-fragment of the ontology in which Instance #798 would inherit the attributes of a Geoloc through its parent City, so that the ground types of Film, Map, etc. would apply to it.

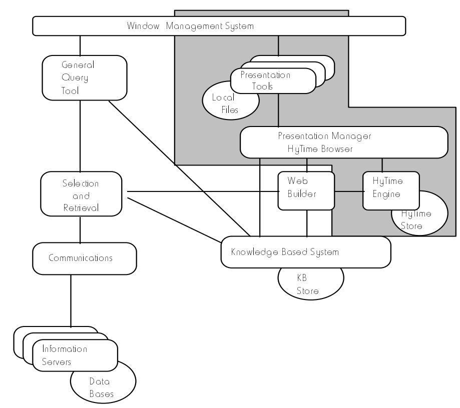

Figure 2 - The architecture of the MIPS system.

Given the requirements for describing each information source that this procedure demands it cannot be economically applied to large number of information sources (e.g. more than 75). Equally, where only a few information sources are available (e.g. less than 10) then it may be less effort to implement and maintain a common data schema between them, or for users to query each in sequence and integrate the results themselves. Consequently, the MIPS approach of mapping semantic based queries into sets of locations based ones, and then integrating the responses is only applicable when between 10 and 75 information sources are regularly used.

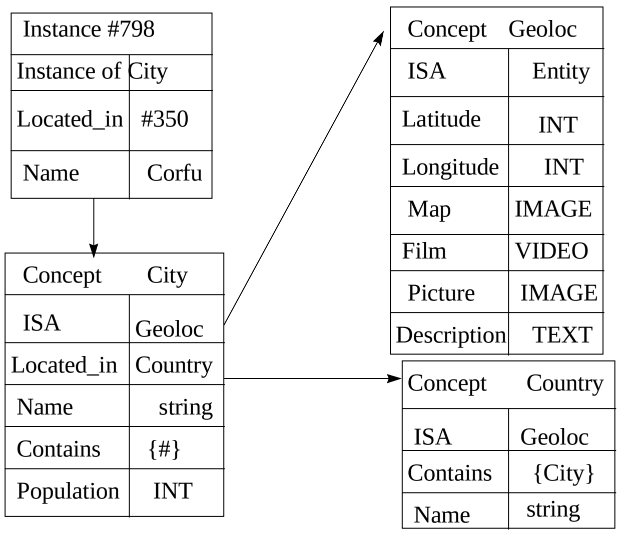

Figure 3 - A pseudo-fragment of the MIPS ontology for the tourism domain

The complete MIPS system is documented elsewhere [4] as are specific components [5,6] and the HyTime standard used to represent the hypermedia web [11] The next section describes in more detail the interaction of the selection & retrieval component and the KBS process to expand the user's query into a set of queries aimed at each target information source and to integrate the set of replies.

INFORMATION RETRIEVAL IN MIPS

MIPS implements each of the eight stages of the information retrieval process listed in the introduction, drawing on both the domain model and user models to support inferences. The generic retrieval process will be described in this section, and the next will discuss the variations that can results from drawing on the user model.

An example of the first stage could be where a user asks for information about a city called Corfu which would be represented as a query in MIPS using the Internal Representation Language (IRL) as :

quant(null,city,X), attr(name,X,Corfu).

stating that their is a city and the name of the city is Corfu. The normal way of asking a question in such query languages is to null quantify a particular variable so that all possible values for it are instantiated rather than evaluated for truth as the variables that are existentially quantified would be. Therefore the query asks for information about a city called "Corfu". In this query, 'city' is a conceptual type, and 'name' is a shortcut for expressing the ground type 'string' which holds the name of the conceptual type. If the query were asking for the names of cities, then a list of strings could be returned from a database, but it is asking about a conceptual type that is not itself represented in any databases and therefore cannot be retrieved. The first step is brought about by the application of the conceptual querying heuristic.

If a conceptual type is queried then it will not be interpreted directly, but it will be expanded into all its attributes and they will be queried. This rules will then be repeated recursively so that if a child of an attribute is itself a conceptual rather than a ground type it will in turn be expanded.

The conclusion of the application of this rule is a large number of ground types which are queried, and which are linked back to the original conceptual types by chains of links (from Figure 3, this includes an IMAGE of a map, VIDEO, TEXT descriptions etc.). This heuristic maximises the number of items to be located, at the cost of retrieval time, access cost or usability of the answer.

The output of the first stage is the largest possible list of ground types that can be inferred from the queried conceptual types in the users request. The second step draws upon descriptions of each of the available external information sources stored in the KBS. Relational database attributes, and tables are described in terms of the domain ontology. For free text retrieval systems and WWW URLs domain ontology terms are used to describe the areas of information available in each source. These descriptions of information sources are matched against the expanded query produced by the first processing stage. Each ground variable in the expanded query can produce one of three results from the match operation: firstly a direct match to one or more information sources; secondly no matches to any information source, or thirdly matches to information sources as a result of inferences to relate the ground variable in the query to the information source description. The most general inference based matches apply the subtype query heuristic; for example, a conceptual subtype inference would accept a database about dogs if the user asked for animals, a geographic part subtype would search a database on Athens when the user asked for Greece. These inferences are directly supported by links in the domain model.*: If an information source description contains a subtype of the variable queried for in the user request then add that information source to the list of candidates for that variable.*

The result of the second stage will be the subset of the expanded query where the ground types match to available information sources, with the corresponding information source location details. This is the level of query which would be produced by the user themselves in an interface which used location rather than content based querying.

This request still contains the maximum amount of location based queries that could be derived from the original content based query. The third stage in the retrieval process is to reduce this set by applying constraints on the retrieval process itself. The first filtering step removes known redundancy by removing any information which occurs in only one sub-query, and is used to retrieve only one data item which is also retrieved from another data source..

The next two rules are applied to reduce the number of candidate information sources according to constraints on retrieval process itself. These constraints are drawn from the user model as described in the next section. These rules apply to the time the retrieval itself will take and the cost of the retrieval itself. Both rules exclude candidate information sources for items where multiple information sources are available and the estimated retrieval time or retrieval cost is greater than the user model constraint. The retrieval time and cost are calculated using metadata about information sources in the KBS detailing the usage costs per retrieval or in time for each information source, along with transmission times and costs.

The fourth general pruning rule on the set of information sources is by the reliability, recency and accuracy of the information available. Metadata on each information source includes ratings of these. Reliability is a measure of the errorfulness of an information source. The recency is a measure of the update rate; for example, a stock exchange quotation source updated every minute has a better recency than one updated only every ten minutes. The accuracy of each data source refers to the accuracy of information stored; for example the accuracy of the measures of stellar distance from the earth vary in different scientific information sources depending on the equipment used to take the measures. Values for each of these rules are retrieved from the user model and applied in the sequence in which they have been described in order to exclude information sources used to locate information which can be retrieved elsewhere . The exclusions only apply where information is available in more than one information source.

Following the reduction in the set of candidate information sources in the third stage of the process, the fourth stage involves the translation of each remaining query into the target language of the external information source, the issuing of the queries to each information source (including clearing security guards for each user), constructing an integrating description of the data expected (the intensional structure) into which it can be placed on retrieval, and placing the retrieved data into this structure. The returned data is then available for the fifth stage of processing where the amount of data returned for each query variable is counted, and further constraints can be applied to it if there is too much.

The sixth stage of processing converts the data into common formats. That is, all dates, currencies, weights, lengths etc. are each converted into the appropriate common format. This is undertaken by applying a simple amount calculus which contains conversion rules for all amounts (values with units) that exist in the returned data. It can be ensured that all possible returned amounts can be converted into standard types since the conceptual schema of all the data sources are described in the KBS, and if new amount types must be added in new conceptual schema then conversion rules to the standard format must also be added to the amount calculus. Complex conversions such as converting costs with or without purchase tax into a common form are not attempted here [13] although the amount calculus mechanisms can also apply such specialised rules if the effort of implementing them can be justified.

Once the data are in standard formats conflicts in the data set can be resolved in the seventh stage of processing (e.g. two different prices for the same hotel). The main heuristic for conflict resolution is to select the value from the more accurate data sources, and select the first value where two data sources of equal accuracy contain the same value. This crude data resolution process should clearly be replaced by a more reliable set of rules, but these have not yet been developed.

The eighth stage of processing is to design a presentation of the retrieved information. MIPS uses hypertext to display data since simple relational tables would not arrange retrieved full texts, or other media in a structured presentation. The hypermedia standard HyTime [11] is used as it permits the separation of the content of a document from the specification of its form and the resulting document should be portable since it is an ISO standard. HyTime templates have been constructed for simple and composite hypermedia objects to which returned data can be bound. The simplest method of designing a presentation is for the user to identify a presentation template in the original query along with the content description, and the processing constraints. However, authors are permitted not to state templates in the query, in which case the KBS will be asked to select or design one on the basis of the content specification in the query and general design rules applied through constraint hierarchies [17]. The design does not address any issues of temporal synchronisation since all retrieved audio and video items are assumed to be complete within themselves. There are only three issues to be resolved by the design: which returned data items should be grouped into single screens (hypermedia nodes with associated presentation mechanisms), what the layout of each screen should be, and what links should be created between nodes (with buttons or hot graphics introduced for interaction). If the structure of the returned data does not match an existing template simple layout planner is used to build a new template and layout the information as described in [6].

The link creation task is resolved simple heuristics linking information about concepts to them, and child concepts in the query to their parents (as in Figure 3). This can then be divided into screens for each concept and then further divisions can be made if simple layout rules and conditions are breached. Richer link structures could be introduced with more heuristics such as :*Link concepts to all occurrences of the concept elsewhere in the existing web or the generated one.* However, such heuristics can generate considerable numbers of links which are not relevant to the user's task [14] so they have not been included at present until they can be controlled by the user model.

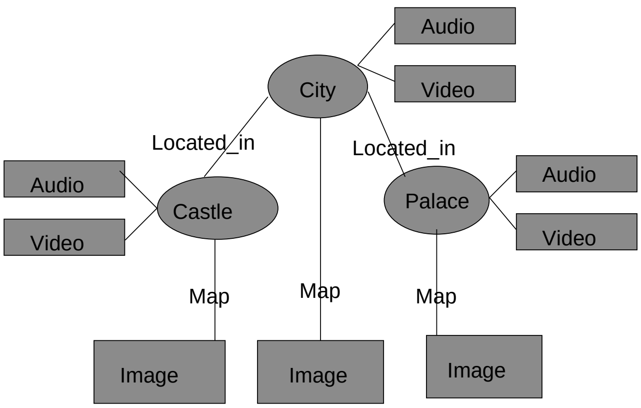

Figure 4 - The web for a query on the city of 'Corfu'.

The query for Corfu shown above resulted in the retrieval of a text description of Corfu, an image, a title for the image, a map, a title for the map, a video with a title for the video, along with two places in Corfu itself (the old castle and the palace) and the same details for each of them including their names (whereas the name Corfu was already in the query). The resulting web is shown in Figure 4 where three nodes are created, each with some text and an image displayed on them and buttons to activate associated audio and video, and to move to an alternative representation as a map. The main node for Corfu also includes a menu of child nodes with two items in it. Each node includes a return button to go back to the previous node and a title showing the name. The presentation of the top node generated for Corfu in this example is shown in Figure 5. In the full demonstration there are 28 sites under the city on the scrolling menu, as well as 36 hotels with information about each of them and the facilities they offer.

USER MODELLING IN MIPS

Following [9] (similar to [15]) the user modelling component consists of a an acquisition interface and a query interface. The acquisition interface allows the assertion of each query the user issues which is decomposed to state the concepts the user is interested in and the preferences the user states for media, processing constraints and presentation organisation in queries. The query interface allows the KBS to inquire about user preferences which apply at each of the processing stages described above in selecting the content of an expanded query, in selecting information sources for those queries, in applying processing constraints to restrict the set of data sources to be queried, and in the design of the presentation. Internally the user modelling component stores a model of the user in terms of the information acquired about the user through the acquisition interface, and in term of a set of user classes. Each class is associated with a set of macro-rules which define a user class, and are applied to the acquired information to determine if a class applies to that user. Once a user is allocated to a class, default values from that class may be used to answer queries of the user model about the user. These default values are derived from micro-rules within each user class which perform the mapping from the user class to the domain class, and the domain generalisations. Users can be assigned to multiple classes and multiple inheritance applies from the classes to an individual user model. Conflicts in domain properties arising from the application of multiple micro-rules are resolved by a simple priority mechanism between classes (the top left class overrides the bottom right on a graph of classes). Evaluation of user class assignment and default inheritance is delayed until query time since considerably more inferences can be drawn about a user than are needed by any single query to the user modelling component.

The method of constructing the user model is to list the options available as query answers and to interview domain (database and presentation design) experts as allocate user classes to each option; define macro rules to trigger assignment of users to each class; and then iteratively reduce the number of classes and macro rules where possible.

This process produces a small set of user classes with rules that determine which users should be allocate to each. The triggering macro-rules can be based on the content, presentation or processing constraints set by a user in a query, or by explicitly asking the user to complete a form when first using the system. For the demonstration domain of Greek tourism where the system is to be used in high street travel agencies, an informal survey suggested a limit of 6 questions should be imposed on this form to ensure completion by users without detracting from their impression of the usability of the system.

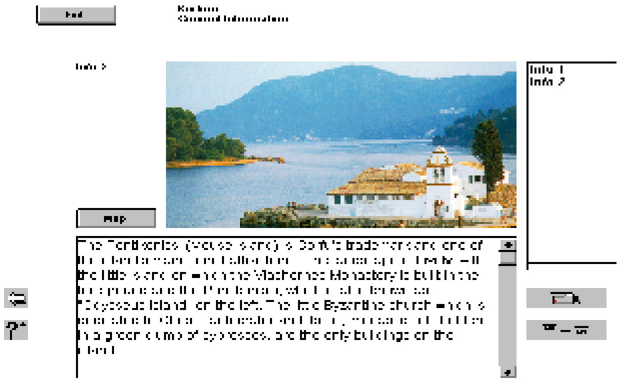

Figure 5 - A screen image for the top node of a web constructed in answer to a query about the city Corfu.

The user modelling component was provided with a programmer interface which shows a graph of the user class network (capturing the conflict resolution order), the macro-rules to trigger class allocation, and the contents of each class (as micro-rules or as copies of queries entered by users). No interface is provided to the end user to explicitly view the user model structure since this would add to the complexity of the system as they perceive it. However, the general query tool through which users enter queries could present current default values of several domain properties (e.g. processing constraint values, presentation options) which are obtained by querying the user modelling component.

The user modelling component can be called at any of the eight stages of information retrieval in order to tailor the retrieval to the user. In the first stage the query is expanded to produce an exhaustive list of candidate concepts to query. The user model can be asked to provide a limited list of links which should be inferred over to reduce this list. For example, may be very specific in their geographic preferences, so geographic proximity inferences should not be drawn. Either general semantic links or domain specific inference rules for query expansion can be excluded in this way. The most commonly used constraint is the natural language of the user which is often used to constrain text and audio media.

In the second stage of processing this list is mapped to the largest possible number of data sources. Again the user model can be called upon to prune this search by limiting the maximum number of data sources to be chosen.

In the third stage the set of data sources is reduced according to constraints of time, cost and reliability of data sources. The thresholds for all three of these are available from the user model.

In the fifth processing stage the amount of returned data is constrained if there is too much. The notion of too much data is entirely dependent on the user. For each media type the user model stores a value for the maximum amount of data to be returned for each concept. This may limit the number of text documents to 10 (a common value used in text retrieval systems) or the number of videos to 3 (for continuous media we have tried both number and total duration as a limit and found number more important since the users can always select whether to play any one item).

The sixth stage of processing converts the data into common formats. Not all formats can be used as a common format since the amount calculus cannot convert every format into every other one. However, if the users explicitly mention any units in a query, the associated family (e.g. Imperial, Metric) can be adopted for presentation. Costs are displayed in the currency associated with the users natural language.

In the seventh stage of processing conflicting data are resolved on the basis of the reliability of data sources stored. Alternative criteria include the trusting of local over remote data sources, data sources from named service suppliers, or named data sources over all others. The rules for these preferences cannot be inferred from user queries, but must be explicitly entered by the user.

In the eighth stage of processing the presentation web is designed. The first set of options here are dependent on the users device and available display tools. MIPS can convert returned media into several formats for display depending on the available tools. The user can set preferences for the tool to use for each media type (in a similar way to WWW browsers).

A second option is the treatment used for absent information that was queried for by the user but there was not available data source for, or for which a data source was queried, but no data was returned. This can be ignored, explicitly reported as missing on a node of its own, or marked as missing in the planned presentation structure.

Although simple, the current design algorithm calls on the user model to resolve conflicts in several design rules ranging from the importance of one medium over another (based on the number of times a medium was asked for by the user) to ordering constraints when there are several nodes at the same level in the web which must be linked (based on the frequency in which concepts have been queried).

CONCLUSION

The purpose of the user model in an information retrieval system is usually to select data as being *like* other data requested. At the other extreme are user models proposed for the construction of multimedia displays which provide the user’s intent to select between design plans [16]. The user model presented here is simple compared to the second since the design process itself is simpler. However, the model is used not only during presentation design, but also during all the other stages of the information retrieval process over heterogeneous distributed data sources. The paper also shows how such a simple modelling component developed for one application [9] can be used in this second with minimal changes.

This paper has tried to present both the design rationale for the MIPS system, and some of the design options for anybody attempting to develop similar systems. MIPS has been demonstrated in a single application to present information about Greek tourism on the island of Corfu. This is a real application for a customer, but it is only one, and that has not yet been evaluated thoroughly.

Several improvements have been mentioned in the paper which have not been adopted yet. It may be that the user model should be drawn on more to change the presentation design or the query expansion. However, the cost of explicitly asking the user for preferences is seen to be high in perceived usability. Unless more macro-rules can be generated to identify the options and triggers within queries it is unlikely that they will be adopted for this application.

ACKNOWLEDGEMENTS

The research reported in this paper was partly funded by CEC Esprit III grant 6542 to the MIPS consortium of Longman Cartermill (UK), Laboratory of the Research Councils (UK), STI (Spain), SEMA Group Benelux (Belgium), Herriot-Watt University (UK), Trinity College Dublin (Eire), DTI (Denmark).

REFERENCES

1. Rich.E. User Modeling via Stereotypes. *Cognitive Science, 3,* (1979), 335-366.

2. Shardanand, U and Maes, P. Social Information Filtering: Algorithms for Automating “Word of Mouth” in *Proc CHI’95,* (Denver CO, May 1995), ACM Press, 210-217.

3. Hurson, A.R., Bright, M.W. and Pakzad, S. (Eds.) *Multidatabase Systems: An Advanced Solution for Global Information Sharing*, (1994) IEEE Computer Society Press: Los Alamitos, Calif.

4. Jeffrey, K.G., Behrendt, W., Macnee, C.A., Wilson, M.D., Kalmus, J.R., and Hutchinson, E.K. A model for heterogenous distributed database systems in *Proc. 12 BNCOD* (UK 1994), Lecture Notes in Computer Science Series 826, Springer Verlag, 221-234.

5. Behrendt, W., Hutchinson, E., Jeffrey, K.G., Macnee, C.A. Wilson, M.D. (1993) Using an Intelligent Agent to Mediate Multibase Information Access. In *Proc. Workshop on Cooperating Knowledge Based Systems*, (Keele UK, Sept 1993), DAKE Press, 27-46.

6. Macnee, C.A., Behrendt, W., Kalmus, J.R.,Jeffrey, K.G., Wilson, M.D., Presenting Dynamically Expandable Hypermedia, *Information and Software Technology* **37***(7)*, (1995) 339-350.

7. Clancey, W.J Heuristic Classification. *Artificial Intelligence, 27,* (1985) 289-350.

8. McTear, M.F. User modelling for adaptive computer systems: a survey of recent developments, *Artificial Intelligence Review, 7*, (August 1993) 157-185.

9. Chappel, H., Wilson, M. & Cahour, B. Engineering User Models to Enhance Multi-Modal Dialogue, in J.A. Larson & C.A. Unger (Eds.) *Engineering for Human-Computer Interaction*. Elsevier Science: Amsterdam, (1992) 297-315.

10. Lenat, D.B. Artificial Intelligence: A crucial storehouse of commonsense knowledge is now taking shape. *Scientific American, 273* (3) (Sept. 1995) 62-64.

11. Newcomb, S.R., Kipp, N.A., and Newcomb, V.T. ‘The HyTime hypermedia/time-based document structuring language’ *Communications of the ACM*, *34*, (1991) 67–83.

12. Arens, Y., Chee, C.Y., Hsu, C-N. and Knoblock, C.A. Retrieving and Integrating Data from Multiple Information Sources. *International Journal of Intelligent an Cooperative Information Systems, 2*(2), (1993) 127-158.

13. Sheth, A.P., and Larsen, J. Federated Database Systems for managing distributed, heterogeneous and autonomous databases, *ACM Computing Surveys, 22*(3) (1990), 183-236.

14. Hill, G.J, Wilkins, R.J. and Hall, W. Open and reconfigurable hypermedia systems: a filter based model. *Hypermedia*, *5*(2), (1993) 103-118.

15. Kobsa, A. and Pohl, W. The User Modelling Shell System BGP-MS. *User Modelling and User-Adaptive Interaction, 4*(2) (1995) 59-106.

16. Csinger, A., Booth, K.S. and Poole, D. AI meets Authoring: User Models for Intelligent Multimedia. *Artificial Intelligence Review*, *8* (1995), 447-468.

17 Borning, A., Freeman-Benson, B., Wilson, M. Constraint Hierarchies. *Lisp and Symbolic Computation, 5* (1992), 223-270.