A Multi-Agent Architecture for Cooperative Query Answering

Michael Barg and Raymond K. Wong
Basser Department of Computer Science
The University of Sydney
NSW 2006, Australia
{mbar, wong}@cs.usyd.edu.au

Abstract

Many people have considered the idea of query relaxation as a means of supplying an approximate answer when an exact match to a database query could not be found. By supplying an answer that nearly matches the original request, it is hoped that the user may still be able to find a satisfactory match. Much of the work to date, however, has not examined the different “relaxation needs” of different users. This paper proposes a solution by using a series of user profiling agents to extract information about the user from their patterns of searching behaviour. Information is extracted by the various agents in a mostly invisible manner, by observing behaviour which the user undertakes as part of their normal searching practice. Multiple agents work in a dynamic manner, obtaining and combining their information as needed by other agents or by the system as a whole. Agents make intelligent inferences about the user and continuously seek and respond to feedback, modifying and refining their assumptions.

1. Introduction

Databases and knowledge base systems are often difficult to use because they do not attempt to cooperate with their users. A database or a knowledge base query system provides literal answers to queries posed to them. Such answers to queries may not always be the best answers. Instead, an answer with extra or alternative information may be more useful and less misleading to a user. To achieve that, a query answering process should collaborate with users to find the information that they are seeking. When users ask queries that do not obtain the information that they seek, a technique called relaxation [8] was proposed to enable a knowledge-based system to work interactively with a user to find alternative answers that are related to the answers of the original query.

For instance, the Cobase group [5, 6] has explored a method for generalizing queries in order to provide related answers. Generalization relies on having explicit hierarchies of predicates and terms over the language. A hierarchy is represented as a partial order. A query rewrite is accomplished by replacing predicates and terms from the query with corresponding predicates and terms higher in the hierarchy. The resulting query is considered to be more general. Such a structure is called the type abstraction hierarchy [4] in Cobase. The hierarchy covers three ways to do ordering: subsumption, composition, and abstraction. For a description of the state-of-the-art of cooperative answering systems and other work on relaxation in cooperative answering systems, read [7, 12] and [8], respectively.

Although relaxation provides relaxed answers to users, it is a general approach to seek additional answers to a query that may or may not be of direct interest of the user. To ensure that the additional answers are relevant, some schemes posed by others for detecting users’ plans and intents may be used to refine the approach. These include the works on analyzing intention [1], user modelling for responding to misconceptions [11], user modelling for guiding query relaxation [2, 3] and generating cooperative answers with respect to users’ goals [14]. In the system FLEX, Motro proposes allowing the user to select directions of relaxation, and thus to indicate which relaxed answers may be of interest [13]. Cuppens and Demolombe suggested the way to identify topics of interest which is compat-
ible with the relaxation approach. Finally Minker and his students developed a relaxation mechanism based on taxonomy clauses in a deductive database [9, 10].

This paper proposes a multi-agent architecture on which we build a user profile for each user, and use this profile in determining which criteria to relax, in which direction (or to which other categories), and by how much. It is also the authors very strong belief that building such a profile must be as unobtrusive to the user as possible. The aim is to build a user profile from information which would be supplied by the user in the normal course of events, asking minimal or no explicit questions. Thus the process of building the profile remains largely an invisible process, which does not require the user to do anything new or different. The system intelligently makes inferences about an appropriate course of relaxation where it has not explicitly observed any queries with the same terms, but has observed queries which are similar. The system also reacts intelligently to feedback regarding the effectiveness its importance determining strategy, re-evaluating and changing its strategy for determining importance so that the resultant importance decision strategy most closely matches the users own behaviour. It is the authors strong belief however, that users should have access to viewing their profile, and possibly even changing it, should they so desire. This profile is eventually used to guide the relaxation so that an optimal set of answers is returned for each specific user.

2. Issues in User Modelling

In attempting to build a system which models human behaviour, there are a number of considerations which must be taken into account. The use of intelligent agents deals with these issues nicely, by reacting and tailoring the behaviour of the system as a whole to each individual user.

The first important issue in building such a system is the recognition that there is no uniformity of meaning for a particular behaviour across all users. If the system assumes that a behaviour on the part of the user definitely indicates one and only one meaning for every user it encounters, the system will fail to be fully effective in accurately modelling each individual user. Thus the system must be able to recognise that different inference rules will be appropriate for different individuals. The agents within the behaviour observation and feedback unit pass information about each user to both the user profile building and inference units, where the strategy employed for each individual is reviewed and refined.

The second very important factor when dealing with any user modelling is the realisation that users are not static entities to be modelled. Users are themselves dynamic and change over time. If one approaches the task of user modelling from the assumption that a user is a static entity to be modelled, it is felt that this approach is likely to fail in the long run. People change. Their preferences change. Their behaviours change. If a system cannot cope with this, then it cannot accurately model the user over a period of time. It is vital that the system be able to recognise and deal with this fact when attempting to model any individual.

3. System Overview

In order to achieve such a system, a complex interaction of a large number of different agents is required. These agents must handle everything from observing the behaviour to building the profile to making inferences to modifying their own behaviour in response to feedback as to the effectiveness of their particular strategy.

The system is broken down into a number of conceptual units, each of which is comprised of many agents which are either intelligent in themselves or exhibit intelligence as an emergent property of the unit. There is constant communication and feedback between agents within these units, such feedback determining each agents behaviour based on feedback as to the effectiveness of its performance to date, giving rise to the emergent intelligence of the system as a whole.

Figure 1 shows the system with all the main agents included, as well as the communication channels employed by each of the main agents. The full names of all the agents shown in Figure 1 are given below. Each of these agents is explained in detail in its own section of this paper.

- Query Relaxation Unit
  QRM Query Relaxation Manager
- User Profile Unit
  - Importance Profile Unit
  IDM Importance Determination Manager
It is important to note that the agents within each conceptual unit are in constant communication with one another. They react and adapt their behaviour based on feedback from other agents so as to behave in the most appropriate manner for each individual user. This is a key point of the system.

4. Information Sources

Given the philosophical belief of the author that the user modelling should be as unobtrusive as possible, there are a limited number of potential sources of information regarding the user.

These are:
5. Query Relaxation Unit

The query relaxation unit has many small agents which carry out the various tasks. The main agent which coordinates all this work is the Query Relaxation Manager.

5.1. Query Relaxation Manager

The Query Relaxation Manager (QRM) is the agent responsible for initiating and carrying out the query relaxation request. The process that occurs is described as follows.

The user asks a query for which there is no answer. The DBMS then passes this query to the QRM, which concludes that it must find and provide acceptable relaxations for this query. The QRM then sends a request to the various agents informing them of which user issued the request (and hence which profile to use) and the query that was issued which had no answer. The various agents then determine the appropriate relaxation information (this process is described in detail below) and passes back to the QRM information on the relative importance of each term as well as the relaxation bands and associated values for each attribute to be relaxed within the query.

The QRM uses the importance information to decide which term to relax first (terms are relaxed in order from least important to most important). It uses CP information to determine which values to relax a query to. An acceptability band is the range of values for a given level of acceptability for a given user. Thus, consider a user for whom a price relaxation of +0-10,000 is very acceptable, a relaxation of +10,001-20,000 mildly acceptable and a relaxation of >30,000 totally unacceptable. The acceptability bands and associated values are therefore \{(1, +0-10,000), (42, +10,001-20,000), (83, +20,001-30,000), (100, +30,001-MAXINT)\}. The first number in each element of this set is an indication of the acceptability level. A means very acceptable, 100 means completely unacceptable. Numbers in between indicate where upon this continuum the associated values lie. Numeric values will tend to look similar to the example, with the acceptability value followed by the range. For discrete values, each element may have more than one value for each acceptability level. Thus an element of an acceptability band for relaxations from the colour “red” may look something like this: (1, “black”, “pink”, “mauve”, “salmon”), indicating that these four colours are all very acceptable relaxations to the colour “red”.

Once the QRM has received the relaxation information, it then proceeds to issue the relevant requests to the underlying DBMS, using the information just received to ensure that the queries are optimally relaxed for the individual user who issued the query. This is achieved by replacing predicates in the original query with predicates implied by the relaxation information obtained.

6. Behaviour Observation Unit

The Behaviour Observation Unit (BOU) is the simplest unit in the system. The arrangement of the agents within this unit is shown in Figure 1.

There are two main agents in this unit - the history manager and the behaviour recorder.

6.1. History Manager

The history manager is responsible for recording the history of all actions of a particular user. Specifically this means recording all queries made by a user, selections made by a user, and where these selections appear in the answer set, both serial position and relaxation band. A relaxation band refers to the group of answers returned in response to a single relaxation. It is necessary to record all this information for instances when
any of the appropriate managers decide that they have not been using the correct heuristic for interpreting the users behaviour. When this occurs, the managers instruct the appropriate agents to use a different heuristic for interpreting the users behaviour. The history is then needed as all previous behaviours must now be interpreted in light of the new heuristic. As well as recording the above data, the history manager also stores the appropriate metadata associated with each observable behaviour. For example, each user query also records the time, which is one of the determinants used by the session determination manager in order to decide when a session begins and ends. When recording a choice which was found acceptable, the metadata includes the exact criteria which would give rise to this specific answer, so that deviation from the original criteria can be calculated.

6.2. Behaviour Recorder

The behaviour recorder is responsible for recording all the appropriate behaviour and sending it to the appropriate agents. All information is passed to the history recorder. Each datum is also passed to the other agents which require it, along with the relevant metadata.

7. User Profile Unit

The User Profile Unit is comprised of two sub-units – the Importance Profile Unit and the CP Profile Unit.

7.1. Importance Profile Unit

The Importance Profile Unit (IPU) comprises of four agents, as shown in figure 1.

7.1.1 Importance Determination Manager

The importance determination manager (IDM) is responsible for providing an overall picture of the relative importance of a series of terms to the query relaxation manager. Thus, for example, if a user asks:

SELECT flight_no, carrier WHERE date='4th May 1999' and origin='New York' and destination='London' and time < '13:00'

It is the responsibility of the IDM to inform the query relaxation manager which of the 4 terms in the WHERE clause, origin, destination, date and time are more important than the others. The IDM sends an ordered list to the query relaxation manager. Terms appearing closer to the keyword WHERE are initially considered more important than those appearing later. This heuristic is subject to revision from the importance strategy manager.

When it receives a request to provide importance information, the IDM queries both the select importance manager and the where importance manager. The IDM considers the answers from both of the queried managers, and uses heuristics to combine the information into a single picture of importance. This is passed on the query relaxation manager.

The IDM uses various heuristics to combine information from both the select importance manager and the where importance manager into a single, cohesive list. As with all other managers in the system, the IDM is aware that no single heuristic will be appropriate for all users. Thus it continually takes note of feedback regarding the positions of acceptable choices (both serial position and relaxation band), and uses this as the basis for refining its own heuristic for combining the two disparate sources of knowledge. The IDM works on the assumption that if the first choice a user makes is quite late in the delivered answer set, then the IDM has followed an incorrect heuristic for determining importance. The aim is to have selected choices from the answer set appear early in the answer set, as this is likely to indicate that acceptable choices were presented early on to the user. The IDM also takes note of clusters of selections. It may happen that a single selection was made in the first relaxation band (say answer no 4), and then the next acceptable choices were in the second relaxation band (say answers 25, 26, 29, 31, 32). The fact that there is a cluster of acceptable answers in the second relaxation band indicates that overall this was found to be more useful to the user than the first relaxation band. This suggests the possibility that the choice of importance was once again not optimal or fully accurate for this user.

7.1.2 Importance Strategy Manager

The Importance Strategy Manager (ISM) is again used to ensure that the heuristics used are appropriate to the specific individual, rather than a general population statistics that may or may not be appropriate. The where and select importance managers use certain heuristics to decide what importance information
is imparted by each SQL statement, and how it is to be interpreted. Initially, before the system has any familiarity with a specific user, the heuristic used is that words appearing earlier in a list of terms are more important than those appearing later.

This heuristic, however, does not take account of different language patterns in non-English speakers or people for whom English is not a first language, nor does it take into account individual differences between users. To deal with this, the importance strategy manager is used to review the way in which the select and where importance managers interpret information contained in each SQL statement with regards to what it tells of importance for that particular user. The ISM has two main mechanisms for evaluating how successful a given heuristic is for a specific individual.

The first involves requesting an inference of importance for a term where the importance is already known. This information is requested from the importance inference manager. The inferred importance is then evaluated in comparison to the already "known" importance. If these two values deviate substantially, the ISM will try using different heuristics on the original data, and see if any of these heuristics yield a result which is closer to that which the importance inference manager has inferred from existing data. In this way, the system can re-evaluate its importance determination strategy and choose on which is the most appropriate for the specific individual.

The second method of obtaining feedback is to examine the serial position and relaxation band of acceptable answers. This uses the same method as the IDM with respect to this information.

Combining this information, the ISM can continually ensure that it is using the appropriate heuristic, and can modify this heuristic intelligently to ensure that the modelling method used is the most appropriate for each individual user. This attempts to ensure the greatest accuracy of the model built for each user.

Whatever heuristic is used, the system does not place too much emphasis on any one instance, instead looking at patterns over time. This is because it is likely that at certain times a user may change the order of importance as implied in their statement for reasons other than a change in their overall perception of importance. Thus it is important to allow for these anomalous instances. The system must be able to distinguish between an anomalous indication of different importance between two terms and a change that signifies an actual change in the mind of the user.

7.1.3 Where Importance Manager

The where importance manager is responsible for using the heuristic dictated by the ISM to interpret information regarding importance contained in the WHERE clause of the SQL statement. It makes use of this heuristic to both extract the information and store it in an appropriate and efficient data structure, with considerations for both space and access time minimization.

7.1.4 Select Importance Manager

The select importance manager performs exactly the same job as the where importance manager, but deals with information extracted from the SELECT clause of the SQL statement. The same considerations as used in the where importance manager are employed here.

7.2. CP Profile Unit

The CP Profile Unit (CPPU) performs the same job as the Importance Profile Unit, but with respect to information regarding Criteria Preference (CP). The agents within the CPPU and their relationship can be seen in figure 1. Note that the CPPU does not have a CP strategy Manager. This is because the heuristics used to determine Criteria preference are considered to be universal. Account is taken in both heuristics for anomalous or aberrant behaviour. These heuristics are described in more detail in the relevant sections below. The heuristics used to determine the way in which the information is collected and the selection CP manager is combined is not static, and is constantly observed and updated as the system learns more about the individual user.

7.2.1 CP Determination Manager

Similarly to the Importance Determination Manager, the CP Determination Manager (CPDM) is responsible for delivering an overall picture of criteria preference to the query relaxation manager. The CPDM acts as the gateway through which the query relaxation manager obtains information about criteria preference. The query relaxation manager requests relaxation information for one attribute, giving the context of that
attribute at the same time. The CPDM then queries the autorelaxation CP manager and the selection CP manager, to see if they have any information relating to the appropriate attribute. If they do, the information is passed back to the query relaxation manager and is also passed on to the CP inference manager as an added datum for use in later inference calculations. If they do not, the CPDM sends a request to the CP inference manager, which looks to see whether it can infer the behaviour appropriate to this attribute from any similar behaviour it has observed from the user. The CP inference manager will pass any such inferences back to the CPDM. Once the CPDM has obtained the appropriate information (however it is obtained), it passes that information on to the query relaxation manager.

The CPDM is responsible for delivering information about the various acceptability bands for a given attribute for a particular user. The job of the CPDM is first to decide what are the appropriate acceptability bands to consider. Then the CPDM must give some measure or weighting to the acceptability of these bands.

If both the autorelaxation CP manager and the selection CP manager do contain information about the appropriate attribute in the required context, the CPDM must decide how to combine the information to form a global picture of criteria preference. As has already been noted, a user may select a choice for reasons other than its acceptability to them (perhaps through simple curiosity). In contrast, autorelaxation is the system observing the user themselves performing the relaxation. As this behaviour has been directly observed in the user, it is considered that this behaviour was at least acceptable in one instance. Thus it is considered more likely that information contained in the selection CP manager may be less reliable than that contained in the autorelaxation CP manager. Thus, the initial heuristic assumes that information obtained from via autorelaxation patterns are more meaningful than those found through observing selection of choices.

This heuristic, however, is not rigid, and can be modified over time in response to observation of the users behaviour. The CPDM observes the actual choices selected by the user, as well as the serial position of these choices (ie which choice was made first, which was made second, etc). It then compares the criteria which give rise to the chosen answer against answers from both the autorelaxation CP manager and the selection CP manager. If the actual selection chosen fits the results of one answer rather than another, then the weighting is changed to reflect this.

### 7.2.2 Autorelaxation CP Manager

Autorelaxation refers to the process of observing a users own relaxation of criteria over time. The important thing about determining autorelaxation is to determine the beginning and end of a single session. A session is the time during which an individual is searching for a particular piece of information. The job of determining the beginning and end of a session are given to the session determination manager (in the feedback unit), which informs the Autorelaxation CP Manager (ACPM) when a session has commenced and finished.

Within a single session, if a user enters a query and does not find what they are looking for, they may enter a subsequent query for a number of reasons:

1. Their first query gave too many matches and they are trying to limit the number of responses.
2. Their first query did not give any/ enough matches and so they are trying to broaden the scope of their inquiry.
3. They are trying to change to focus of their query, thus broadening some parts and restricting others
4. They are searching for an entirely different piece of information altogether.

In case 2 above, and in a portion of case 3, the user is performing their own relaxation. This is referred to as autorelaxation.

Within a single session, whenever the ACPM encounters a new attribute, the range of acceptable answers for that attribute is considered to be the baseline state for that attribute. Baseline information is often encountered in the first query of a session, but not always, as the user may introduce new attributes into the WHERE clause. Consider, for example, the query:

```
SELECT vehicle WHERE type='car' and price<20,000 and colour='red'
```

The baseline states for the attributes type, price and colour are 'car', <20,000 and 'red' respectively. Note that some of the baseline state consist of a single value ('car' and 'red') whereas some consist of a range (<20,000).
A subsequent query is said to be autorelaxed if the range of acceptable answers for a specific attribute is either greater than the range in the baseline state, or the same size but a different value. Consider that the same user now makes the following query:

```
SELECT vehicle WHERE type='car' and price<30,000 and colour='black'
```

In this query, `type` has not been autorelaxed (it has the same value as it did in the baseline state). `price` has been autorelaxed, as the range <30,000 is greater than the range <20,000. `colour` is also said to be autorelaxed. The range is the same (ie it only consists of one acceptable value) but that value is different from the one in the baseline state.

The ACPM keeps track of all relaxations and the contexts in which they occurred. Recording a relaxation involves recording not only the relaxed value, but what it was relaxed from. Thus, in the above example, it is not recorded that `black` is an acceptable relaxation of `colour`. Instead, it is recorded that `black` is an acceptable relaxation from `red` for the `colour` attribute. The job of the ACPM is to keep track of all relaxations and associated meta data in a way that minimises both the storage space and the access time for the relevant data. When determining the acceptability bands the ACPM uses the relative frequency of observable acceptable deviations as the basis for determining how many bands to return to the CPDM.

### 7.2.3 Selection CP Manager

The selection CP Manager (SCPM) is responsible for keeping track of all information inferred from answers presented to the user which were selected by them. This method uses the heuristic that selecting an answer is an indication that it is likely that the user found that answer acceptable, at least enough to want to find out more information about it. However, this heuristic is not infallible. Consider someone searching for restaurants where the main meal is less than $40. Consider that after being relaxed, on of the answers they are presented with is "Betty Blue’s Beautiful Bountiful Beneficial Banquets", which has main meals starting from $100. The user may select this item purely out of curiosity aroused by the intriguing name. The fact that they selected this item may indicate nothing whatsoever about the acceptability of main meal price of $100. The SCPM must not only record all information derived from selections made, but it must attempt to distinguish selections which were made for reasons other than acceptability. It assumes that if a particular deviation has occurred infrequently relative to the amount of information contained for that attribute, that it probably indicates a non-acceptability choice, and thus excludes it from any calculations of acceptable relaxations.

Criteria preference information is extracted from selections by comparing the criteria which give rise to the selected answer as compared to the criteria specified in the original request.

Consider the example:

```
SELECT vehicle WHERE type='car' and price<30,000 and colour='red'
```

Suppose the user selects an answer which has the associated attributes for `type`, `price` and `colour` as 'car', 26,000 and `black` respectively. The attributes of both `type` and `price` do not deviate at all from the original criteria specified, as they exactly satisfy the boolean conditions entailed by the relative criteria. `colour` however does display a deviation from the specified criteria of `red` to the actual value of `black`.

As with the ACPM, the job of the SCPM is to record all acceptable relaxations again with the aim of minimising both space and access time. The SCPM uses the same mechanism as the ACPM when determining acceptability bands.

### 8. Inference Unit

The inference unit is comprised of only two agents, but the work of these agents is extremely complex and intricate. It is within these two agents that much of the most intelligent reasoning takes place. Both inference managers attempt to make inferences about behaviour based on situations in which the behaviour is similar. The CP inference manager makes extensive use of the Logical Abstraction Hierarchy (LAH). Full details of the LAH can be found in [3]. A summary is presented below.

#### 8.1. Importance Inference Manager

The main role of the Importance Inference Manager (IIM) is in inferring importance to be used as a feedback measure for the Importance Strategy Manager. The IIM is sent a request for its inference of the importance relationship between two terms. the IIM will
then infer the importance relationship between these two terms and give a certainty value which indicates how certain the IIM is that its inference is correct. This inference is then compared against the actually occurring value in the Importance Strategy Manager (see the section "Importance Strategy Manager" for more details).

The IIM uses two different heuristics to infer importance information for the user. The first is by observing the relaxation band in which the selections made by the user occurs. The IIM looks at these with regards to the term that has been relaxed, rather than the relaxation band itself.

The second heuristic for inferring importance is based on autorelaxation information. By observing the order in which the user themselves relaxes various terms, the system has one more independent means of identifying importance information. This is considered to be a fairly reliable avenue of importance information, as it is directly observing the users own behaviour. Once again, the system must realise that it is possible that the user will engage in behaviour which is not indicative of their normal behaviour, and thus it must cope with erroneous or misleading behaviour. However, this is considered a fairly reliable method of independently obtaining importance information.

8.2. CP Inference Manager

The CP inference manager (CPIM) is used to infer a relaxation profile for a particular attribute in circumstances where the system has not had any (or much) previous experience with this particular user with regards to this attribute.

The CPIM makes extensive use of the LAH as the means of inferring a particular attribute profile. This process is described in detail in [3]. The LAH uses two measures of similarity. The first is logical proximity, which indicates how logically similar different contexts are. The other is a measure of how representative of the general case a particular profile is.

The CPIM uses a variety of heuristics to combine information from these two sources. Again, the system uses feedback from the actual selections made by the individual to revise and refine the heuristics used so that the heuristic is specifically appropriate for each individual user.

9. Feedback Unit

The Feedback unit contains many small and interdependent agents. However, the agent with the most complex and interesting job is the session determination manager.

9.1. Session Determination Manager

A session is defined in this system as the period of time when an individual is searching for a particular piece of information, with an upper bound of one sitting at the computer. Thus if the individual logs out, the session is considered to be over, irrespective of whether they found the information they were looking for. This means that if someone logs in to the computer, searches for information on dalmatians, then searches for information on yachts, then searches for information on quasars, they have had three sessions during their single sitting at the computer. Session determination is a key part of the system. It is also extremely complex and difficult to accurately determine.

Many metrics exist to attempt to determine the termination of a session. None of them, however, are absolute, and all are subject to causes other than the end of a session. Metrics include time between queries, term similarity with previous queries, explicit questioning of the user and overlap of answer sets.

10. Summary

This paper describes a system which extensively uses agent technology to arrive at an intelligent, adaptive system which responds to the user, learns and modifies its own behaviour according to the effectiveness of its strategies. The system aims to model the user with regards to query relaxation. It contains a recognition that no heuristic is appropriate for everyone, that people do not always act in logical or consistent manners and that people change over time. In recognition of this fact the system employs many agents to not only model the user, but also to examine and modify its own assumptions about how it obtains the information regarding the user. Thus the system learns what strategies are appropriate for each individual user and is continually evaluating and changing its own actions to ensure that the best possible result is obtained for each individual user. It is hoped that by having such
an adaptive and intelligent system, the user will obtain a set of answers which is optimal to them as an individual.

References


