Goal Oriented Queries: the next level in effective Query Management

The success of the client/server revolution in delivering tools for data access to the desktops of business managers is fast becoming a MIS nightmare of repeated, poorly conceived queries exhausting network and technical support resources.

The cause of the problem is simple - querying a huge database for a particular business purpose is to thread the proverbial needle. The form of standard query languages results in very brittle record sets. If the scope of the query is too wide, an avalanche of information is returned. If the scope of the query is a hair's breadth too narrow, no information is returned. Fine tuning of queries to business objectives is therefore critical to efficient business practice, yet present querying environments provide little or no support for this.

The business goal/standard query language gap

Generally speaking, the business user turns to searches in the pursuit of a specific business goal, whilst the design and nomenclature of the target data resource reflects technical record-based considerations. Data driven queries of the form [Create Date >= Jan 1st, 1997 AND Customer's Name Is Like "Smith"] do not easily reflect business thinking or objectives, such as identify "the best" sales prospects. Standard query languages tend to be exclusive, absolute (Boolean) and quantitative in nature, whilst the expression of business objectives tend to be inclusive, relative and often includes qualitative considerations.

When a user finds that a standard query returns 100,000 records based on what they felt were reasonable guesses as to minimum conditions that likely candidates should meet, but they had in mind to find the "best" 2000, they are inclined to tinker with the criteria cut-off values to better filter the records. Not only is the resultant trial and error approach resource expensive, there is often no direct evidence that the resultant smaller record sets will better meet the original business goal.

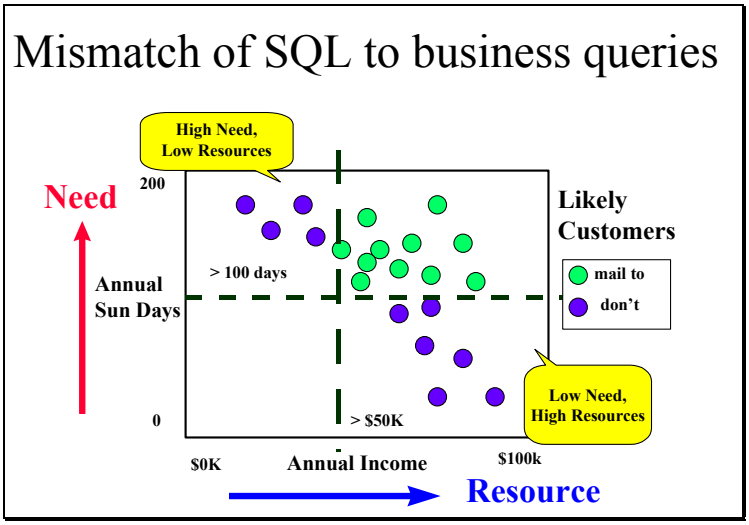
Even if a usable number of queries is returned, the business user has to be very nervous as to whether those returned really match the original business goal. Little or no support is provided to help a user validate the returned record sets.

Repeated attempts by the business user to hit upon the standard query that returns a usable number of appropriate records results in large overuse of data servers, network and technical support personnel. As the number of records in business databases goes from millions (large databases) to hundreds of millions (massive databases), these problems will compromise business competitiveness.

Author: Philip Murphy, CEO, InfoHarvest Inc.
Updated: 8/06/2002

Unsuitability of Standard Queries for Business Queries: scenario

The basis of customers likelihood to buy a product or service is the trade off between need and resources. For instance, a person selling an expensive total sun block shampoo might decide that people with $50k or higher income would have the means to buy their product, and people living in states with more than 100 days sunlight a year would have the need. Common sense tells you that using those values as cutoffs would eliminate many potential buyers - rich folk who buy though their need is not great, and poorer people who live in the arctic where the need forces them to buy at what ever costs. If a weighted query was set up, these would be captured properly by an algorithm of the form S = Sun Days + Income. The inability of standard queries to capture tradeoffs rather than enforce cutoffs ((Days > 150) AND (income > $50k)) forces them to either include customers that aren't good prospects, or exclude some that would be.



Prioritized Query returns

Many business-based information searches lend themselves to prioritized query returns - i.e., a sales representative looking for the *top* 200 sales leads, a lawyer looking for the *closest* match to a present case, etc. By implementing these comparative metrics as formulae that act on each record to provide an overall score of "goodness of fit", returned query sets may be sorted by that score. Instead of 20,000 equal returned records, the user has a prioritized list, and can skim the "best" returns based on that prioritization.

The simplest implementation of a prioritized query is as a weighted query. The user still constructs standard query clauses, [Create Date >= Jan 1st, 1997] but also assigns a "weight" to each such clause, often on some arbitrary scale for 1-10. Each record is evaluated as to whether it satisfies the each *separate* clause, and if so, the weight for that clause is added to that record's overall score.

One of the most important advantages of this approach is that even when a record fails some search clauses, but satisfy other more important clauses, they will still receive a high priority amongst the returned records.

Another key advantage of prioritized queries, is that they can be executed in parallel, on single or distributed databases in a very efficient manner. If the querier only requires the top 2,000 records, and the data is distributed as complete records over a number of sites, the top 2,000 can be determined at each site, compared against those on other sites, and using the decision score as an index, quickly filtered to provide the top 2,000 records across all sites.

However, weighted queries suffer from two critical flaws that make them highly unsuitable for massive searches.

1. They are still Boolean in terms of the individual fields
2. There is no support for validating the choice of weights

1. The individual standard query clauses are still Boolean
Standard query clauses are Boolean - each record either satisfies it or it does not. The business world is one of continuous degrees. If a clause reads as [1996 Earnings >= $50m], is a company earning $49.9m worthless in this respect? Obviously not. This type of aberration can highly distort results, and leaves the user with the problem of guessing the perfect cut-off value.

2. No Weight Validation
Should the business manager weight the clause [Create Date >= Jan 1st, 1997] twice as important as [Customer's City = "Seattle"] ? Can they explain to their audit group why they did so and as a consequence mailed over 50% of their marketing materials to rural areas in King County? When the query results sets are examined it is often not obvious how, if they are found to be unacceptable, the weights should be adjusted to better capture the business intent.

This lack of validation is critical, as it effectively restricts weighted queries to only a few clauses as the user quickly looses control as the number of weights increase.

The science of MultiCriteria Decision Analysis

Decision analysis is a branch of operations research and management sciences that for over 100 years has concerned itself with, amongst other issues, understanding how weights should be applied in what is called MultiCriteria decision making.

The key aspect of standard multicriteria decision making is that it concerns itself from the outset with formulating appropriate scales to measure criteria, to understand the impact of weighting criteria on decision outcomes, and the handling of uncertain and qualitative information.

In a typical approach, a decision goal is established, e.g., "select the best candidate for the position of Director of Sales", and criteria by which candidates will be judged are identified. Scales by which candidates may be measured against those criteria are established, and the importance of the criteria set (as weights). A weighted decision score is then calculated for each candidate, a higher score indicating a better fit to the overall goal. The candidate with the highest decision score would be the suggested choice.

Various analyses to establish how reasonable is the outcome and how sensitive that choice is to the model weights and scales have been well established. Tradeoffs analysis is becoming a very effective way of availing of the author's knowledge of the business world to evaluate the reasonableness of the weights ascribed in the decision model. If the tradeoffs indicate that $30,000 in price was being traded off against a one ounce reduction in metal content, that may, depending on the business context, indicate that a procurement model is out of whack.

Goal Oriented Queries - the application of decision analysis to query formulation

By exploiting decision analysis for query generation, a new approach to business information exploitation is realized - Goal oriented Queries.

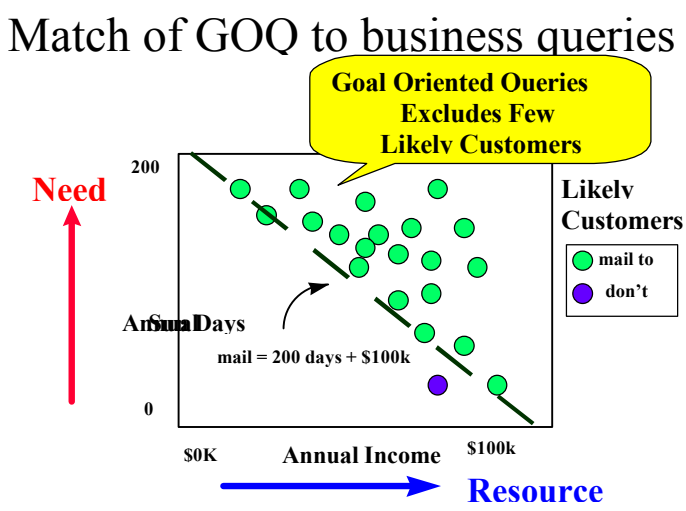


Figure 1: Goal Oriented Queries are suitable for business processes

Goal Oriented Queries are based on weighted sums that effectively trade criteria on which they are based off against each other. Queries based on rules such as "mail to an address if Days/200 + income/100k > 1", are generally much better at discriminating between records when it comes to supporting business activities.

Using the Weighted Decision Object, Criterium DecisionPlus, and an architecture designed by InfoHarvest, all the advantages of decision analysis tools may be combined with database query engines.

Criterium DecisionPlus™

InfoHarvest Inc.'s leading decision management tool, Criterium DecisionPlus, combines a visual user interface, brainstorming, flexible weighting mechanisms, unique analysis capabilities and comprehensive report generation to help decision makers effectively formulate, validate and communicate complex decisions.

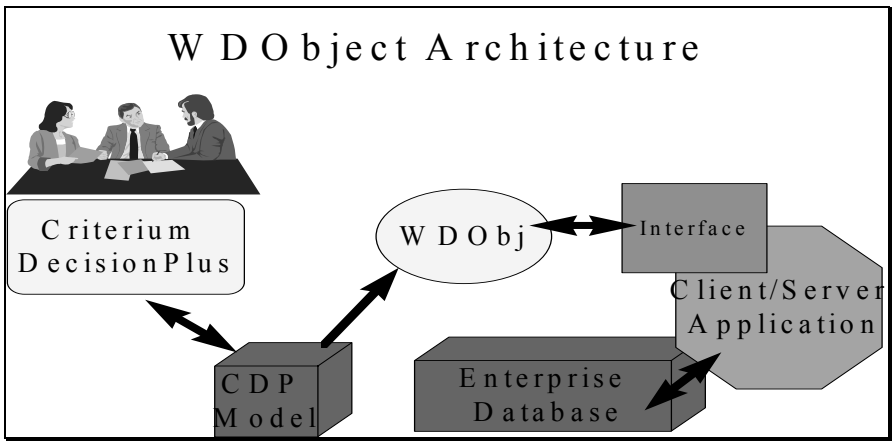
CDP allows decision makers to combine both weighted decision scoring with standard query rules.

1. The decision algorithms within CDP return a decision score for each alternative (record), using the weights and scales determined by the authors.
2. The "Rules" are akin to standard query clauses and are used to exclude alternatives that fail minimum requirements. All rules are evaluated as an AND for each record, and whether each record passed all Rules or not, is recorded for each record.

This visual authoring tool, presently marketed as version 3.0, runs under Windows 95, Windows 98, Windows 2000 and Windows NT running SP 3 or higher. It supports both the Analytical Hierarchy Process and Simple MultiAttribute Rating techniques.

The Weighted Decision Object

InfoHarvest's WDObj product is an OLE object (available as EXE or OCX) with the key decision analysis algorithms used in InfoHarvest's flagship product, Criterium DecisionPlus™. The object has no user interface of its own, instead, its internal data structures and methods for acting on that data are exposed and documented for use by programmers for integration into enterprise critical systems.



Using Criterium DecisionPlus to author their business query, business managers can utilize the WDObj to rate a set of records and evaluate if they satisfy all associated rules. The authors might include a number of sample records in the model they create, but once their Goal Oriented Query is established, the WDObj allows them to access large databases of such records, returning only the most pertinent records. The result is a close tie between the business goal and the returned query sets, a fully documented query, and no need for MIS technical intervention.

Examples of the benefits for

Consultants

An environmental consulting company has worked with a major oil company using CDP to prioritize a number of storage facilities with leakage for waste remediation measures. Criteria such as type of contamination, public exposure, worker safety etc. are employed. The prioritization, once complete is used for planning and budgeting purposes. The oil company can now add this CDP prioritization model to its enterprise information system, so that all of its several hundred sites can be continuously monitored, and prioritized for treatment.

Business Managers

A software VAR has completed a major project for an international bank, imaging and tracking key documents on the trading floor. The VAR would like to use the WDO to include a feature that would analyze the document sets to see which trades are most at peril of non-compliance with SEC rules. CDP would be used to create the model to score each trade in terms of non-compliance. The model would be run on the WDO as part of the VARs solution. When the score exceeds a pre-set threshold, the trading manager would receive an automatic alert indicating that a particular trade might not be properly documented.

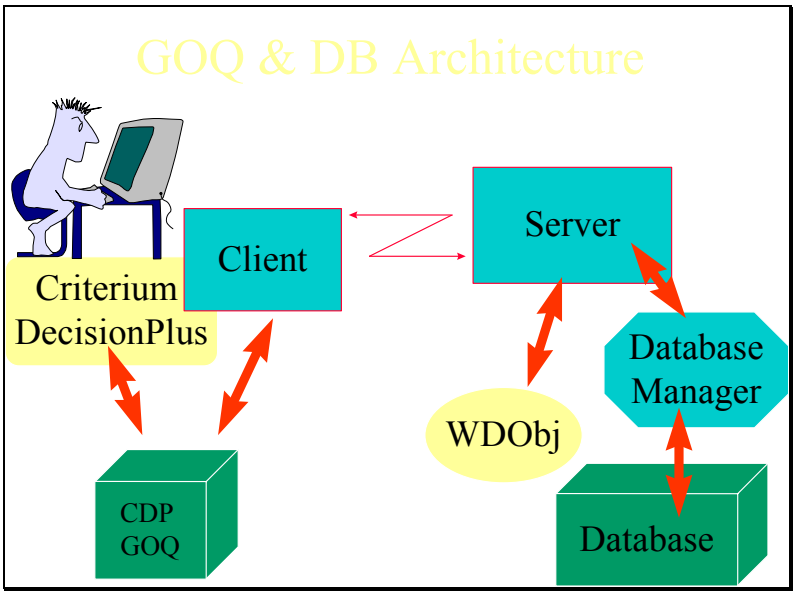
MIS Managers

Your job at a real estate company is to build an application for your Realtors that will return all properties in their region. Since there are thousands of properties in each region, the Realtor wants to be able to order those properties according to what they believe is hot in their area. They want to balance commissions against the difficulty of selling large homes during an economic downturn. At the same time they must allow for the demand on family housing, and the buyer's desire for low taxes. As an MIS manager, you distribute a basic CDP model created by your core analyst to the various real estate offices, they adjust the weights of each criteria to better reflect local conditions. Based on their individualized models, the WDObj ensures they get only the top 200 properties in their region on which to concentrate. Since CDP supports the Realtor in terms they understand, the Realtors can maintain their CDP models themselves.

Implementation within Existing Database Architecture

A Goal Oriented Query in an effective implementation on a client/sever system might have the following features.

1. Using CDP, the business user authors a decision query, concentrating purely on the



   business needs with which they are working. The decision query would be validated against a few example alternatives.
2. The Client facilitates the matching of the user defined criteria and scales to the fields and data types of available databases, creating a Mapping Table. Ambiguities or unavailable criteria are negotiated and resolved with the business users. (The existence of Data Warehouses will greatly facilitate this mapping.)
3. The decision query and mapping table are sent via the Client Application to the database Server, and the desired number of records specified.
4. The Server, via the OBJ, separates the rules clauses and formulates them, based on the mapping table, as a standard query against the database, returning a set of matching records.
5. Again using the WDObj to access the decision query, the Server calculates the decision scores for each record in the Record Set. If available, the algorithms within CDP would be executed as stored procedures within the database. The algorithms may involve table lookups, non-linear functions and text matching.
6. Finally the desired number of top scoring records are returned to the user via the Client application.
7. The GOQuery would be validated by applying CDP's various analysis tools to the prioritized Record Set.

This implementation

- Allows the business users to efficiently determine the information they need
- Minimizes Net work traffic
- Allows the user to fully validate the query results

Integrating GOQ as an extension to the Database engine itself

The long-term aim is to integrate the procedures entailed in evaluating the decision score of a record as an extension to database engine themselves. Providing a mechanism to execute weighted, scale-based queries, whatever the authoring tool, is critical.

Some of the critical aspects involved are:

1. Extending InfoHarvest's Decision Information Protocol (scales, structures, uncertainty)
2. Implementing ordered scales (text and numerical)
3. Implementing effective uncertainty engines

Integration with Other Emerging Information Technologies

Multidimensional Data Bases

Multidimensional databases stores information in structures optimized for advanced query and analysis. The dimensions along which information is organized in these schema corresponds directly to the criteria of decision analysis. All that is lacking from the decision analysis point of view is a group of scale ordering ordering transforms. This is described further in the InfoHarvest White paper "Goal oriented Queries and OLAP".

Data Mining

If the enterprise is fortunate enough to have both a data mining utility and a GOQ front-end both deployed upon the same data mart, the GOQ front-end can use the results of most data mining analyses. By replacing the expert's intuitive sense of the relevant importance of the various fields to the Business Goal by the correlations of those fields gleaned from Data Mining, the business manager benefits from historical data where appropriate. If market conditions shift abruptly, the business expert can use the GOQ model to re-impose their own intuitive analysis when necessary.

Conversely, for new products where prioritization is first performed according to the GOQ model of the business expert, as sales data is accumulated, data mining can then be implemented to validate and improve the intuitive preferences.

Summary

Goal oriented querying provides a more efficient mechanism to retrieve information relevant to a particular business goal. The queries are developed in fashion intuitive to business managers, but effective against he largest databases.
