Web Based Decision Tool

Gaining Decision Superiority

AUGURI CORPORATION

03/05/2008

Authored by: Fadi Victor Micaelian

Table of Contents

Web Based Decision Tool

Gaining Decision Supriority

The Art of Enterprise Decisions

Decision Support is the art of selecting the best solution from a set of choices. While traditional business intelligence tools emphasize data retrieval and summary, they overlook a critical element of the decision process: tradeoffs between criteria. Auguri focuses on the decision process itself, offering an ideal tradeoff-based platform to optimize decisions. Auguri's enterprise class solution spans a wide range of decisions, from simple to complex, from stand-alone to collaborative. Auguri's users can quickly, confidently and successfully optimize their decisions by trading off decision criteria. By optimizing the decision, shortening the decision making process and reducing its cost, Auguri delivers a compelling ROI.

Tradoff Based Decisions

Decisions are based on tradeoffs. A decision is the selection of the optimal alternative(s) from a set of choices. The decision is typically made by trading off -or weighing the relative importance of- a set of factors or criteria. What makes tradeoffs necessary is that usually these factors are conflicting. For example, a consultant that often travels to deliver presentations is interested in a light weight laptop with a large screen. However, the criteria are conflicting; the larger the screen the heavier the laptop. Hence there is a need to tradeoff between conflicting criteria.

The objective of a decision, a selection, a prioritization, or a triage is to select the best solution(s) from a set of alternatives. To be able to make a selection, the first component required is a set of data (the choices). This is the database of alternatives which will be ranked according to their weighted proximity to the ideal.

In order to make a decision we typically use a set of criteria.

These criteria are encapsulated by their *Criteria behavior* (shown on the left side of Figure 1) and typically expressed by a unique function that can be thought of as a utility function, or the way we think about this specific criterion. For example a screen size is a criteria when choosing a laptop, and typically the behavior of this criterion is “the larger the better”. It is important to note that these criteria are often if not always conflicting. In the same laptop selection weight is another criterion. We typically seek lighter laptops. However, the weight of the laptop increases with the size of the screen. Hence we have a conflict between these criteria. Humans have only one way to mitigate these conflicts: TRADEOFFS - a technique our brains have mastered and computers have not been able to handle until the advent of Auguri-.

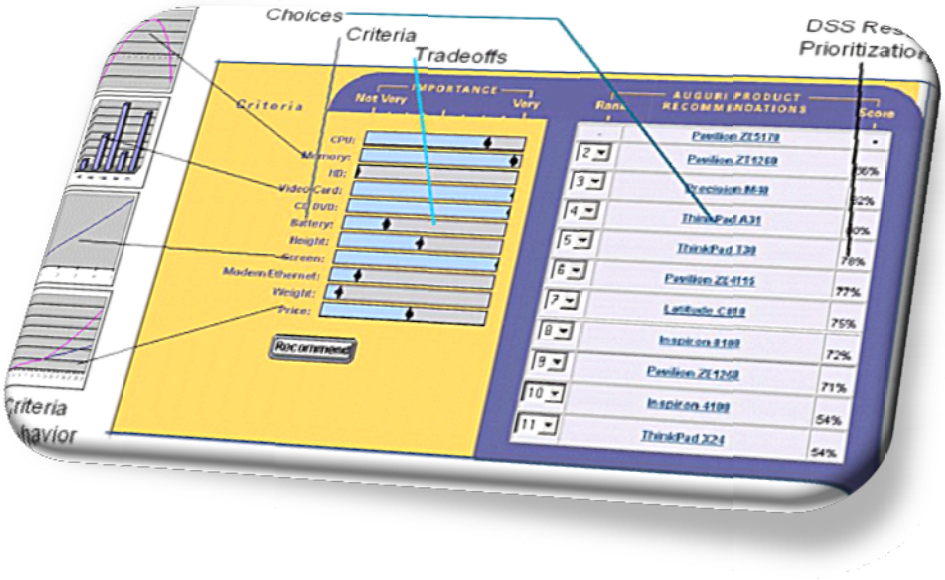

Figure 1: The building blocks

It is important to mention that Auguri offers an extensible library of criteria behavior. The key component of an Auguri search is *tradeoffs*, which model the importance of criteria relative to each other. The tradeoff weights are shown centrally in Figure 4, and ensure the results of the search are highly relevant.

Auguri's decision support system allows the user or to perform the following:

(i) identify and define the set of alternatives
(ii) define the various factors that impact the decision
(iii) determine the way these factors behave and impact the decision
(iv) set the weights of the various factors to reflect their relative importance.

By doing so the decision maker defines an ideal solution in a multivariate system. Each alternative is compared to the ideal solution, leading Auguri to recommend the alternative that is closest to the ideal. Rarely the real world solution will be an exact match with the ideal. Auguri's result is usually a list of alternative with respective scores that measures how far each alternative is from the ideal. This score highlights the relevance of the solution with respect to the decision at stake.

Real Time Decisions

One of the major innovations Auguri brings to the Decision Support technology is the separation of two components of a decision:

(i) the part of the decision that is circumstantial and thus highly dependent on the situation in which the decision is made. This component is altered and modified in real time
(ii) the part that does not change with the circumstances. This is typically an expert know-how.

By separating the part of the decision that is contextual to the situation from the part that is learned and where the expert input is critical, Auguri's human-centered decision support information technologies will improve the performance of less-experienced users. With this innovation, you enable common users to

leverage the expert's know-how while adding their own nuances at decision time, thus making them optimize their decisions as if they were experts, in real time.

Uncertainty

More complex decisions include taking into account information that is associated with various levels of uncertainty. Auguri's Decision Support System is designed to take into account and manage uncertainty. The uncertainty can be either associated with inaccurate or incomplete data sets, or intrinsic to the nature of the decision such as the unknown market response to a price change. The latter is handled by Auguri's domain ordering or criteria behavior while the former is managed at the criteria level and the weights associated with these criteria.

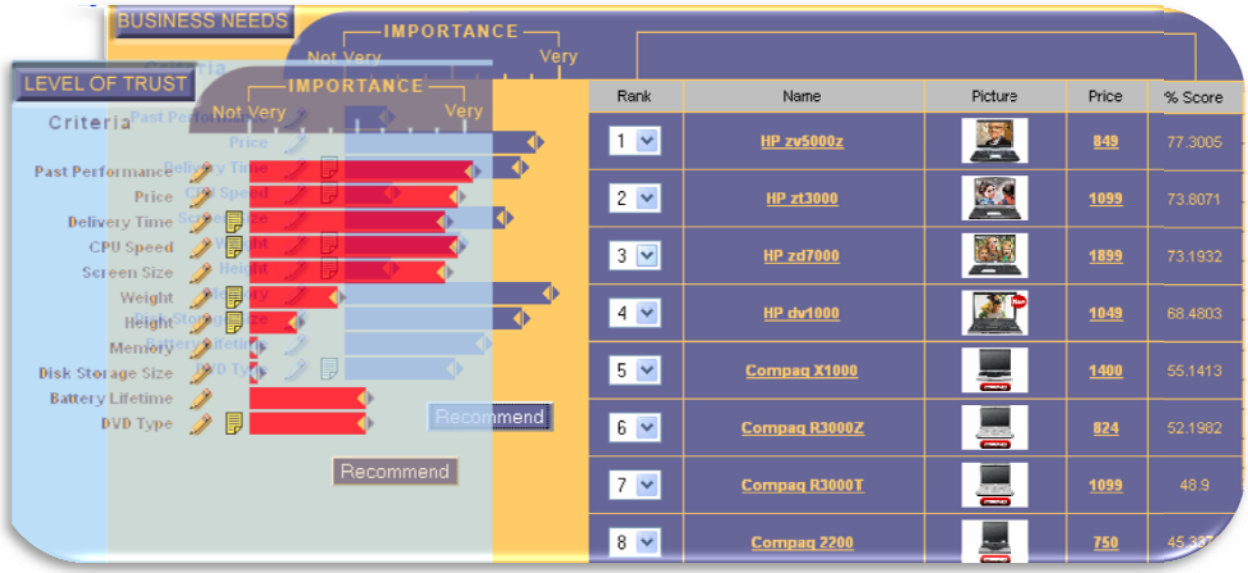

Figure 2: Multi-layer, multivariate collaborative decisions

Like multi-layer chess, Auguri will offer in its next release the ability to have multiple layers of tradeoffs. One layer may be used to model the business decision, and the second may be used to reflect the level of confidence in the data. On the battlefield for example, the course of action choice (attack, retreat, outflank, …) depends on the number of tanks you have, the number of tanks the enemy has among other factors. However, it may be necessary to overlay the decision with another matrix of weights that reflect the level of confidence in the data. So if the satellite is pointing to 10 enemy tanks, your informant confirms that number and you can see 9, your level of confidence is high. While if the satellite count is 102, your informant indicates 2 tanks and you see 9, your level of confidence is low. This would affect your decision on the course of action.

Collaborative Decisions

The purpose of collaborative decision software is to bring the experts in various areas of the business together into a decision that impacts each area of the operation. Because Auguri separates the various factors affecting a decision the system is ideally suited to facilitate the intervention of experts and others in the decision process.

This capability is designed to enable various members of the organization to leverage expert know-how, yet be able to customize the decision to their specific situation by modifying one of several components

of the decision process: the ideal, the criteria behavior, or most often the relative importance of the criteria.

Participating in the process involves (i) Asking the experts for their input in each area (or criterion) that impacts the decision (contribution factor / domain ordering) (ii) Allowing various players to input the importance of the criterion that impacts them and (iii) Allowing a higher level authority that has at stake the overall interest of the project/company to override the weights of the various criteria

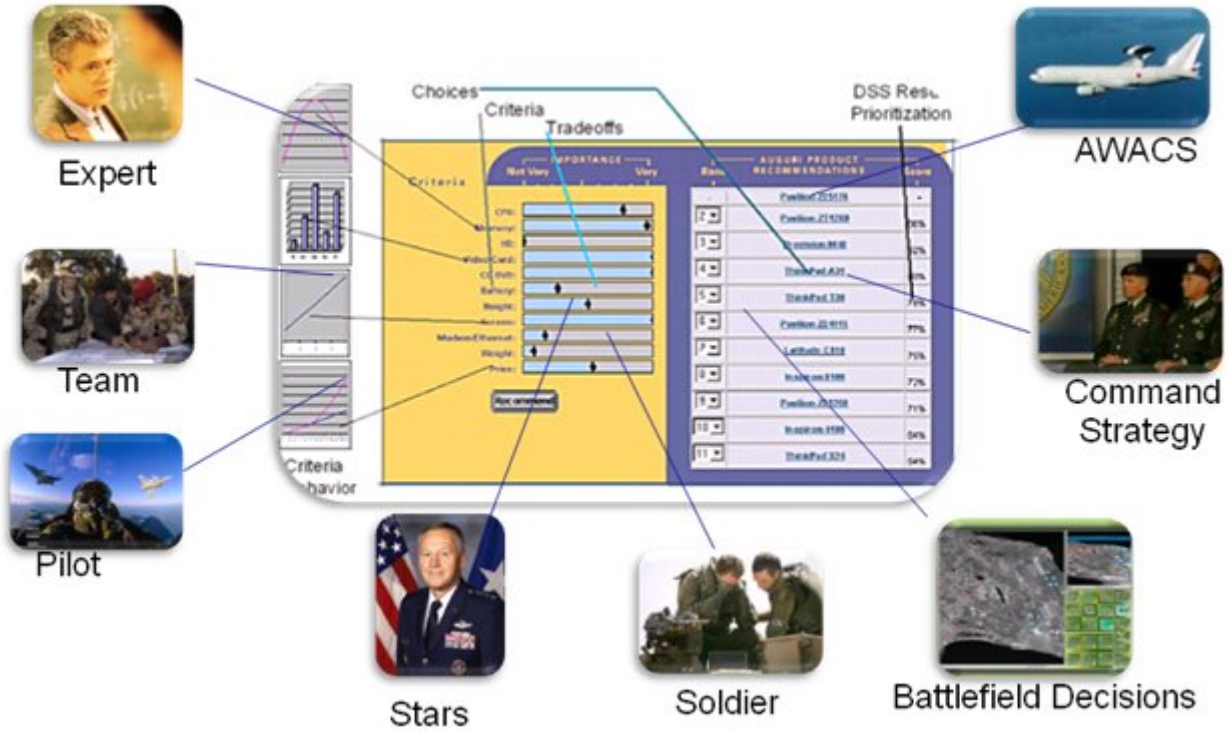

Figure 3: Collaborative enterprise decision

Sharing the experts' know-how, raises the level of competence of the whole organization. Now novices can perform like experts. In addition, by capturing the experts' knowledge in the system, the organization keeps this asset as the experts depart the organization achieving major savings.

Understanding and Justifying a Decision

Decisions rarely follow the same thought process. Sometimes they are made through a rigorous selection process. Occasionally they are made by reference i.e." I like the car that Bob drives, I want something similar". Often they are made intuitively. Auguri's inference technology derives the selection criteria (and their importance) from the results or the selection. With its inference engine Auguri offers an ideal way to understand, rationalize or justify a decision.

Sharing Decisional Intelligence Across the Organization

Auguri's tradeoff based metaphor makes it possible to rationalize, collect, and store insight into user and organizational profiles, preferences, and decision processes, which is not possible with traditional (SQL-based) tools. Auguri savvy applications can now share this knowledge throughout the organization. This raises every member's competence to that of an expert. Furthermore, since knowledge is no longer kept with individuals, it is not lost when they leave. In essence, Auguri-based applications inaugurate the era of intelligence interchange. Another aspect of the intelligence interchange is the ability to capture customer behavior and preference. For example, if you were able to capture the customer preferences and tradeoffs, and download them from the brain of each salesperson to a centralized data-warehouse you would have tremendous intelligence on your customer needs.

Business Intelligence

Auguri gives a totally new perspective to your data. The analytical tool can gather intelligence in the absence of historical or demographic information. Data Mining can be performed without the complexity of traditional tools. With its ad hoc querying tool, it puts corporate intelligence at the fingertips of lines of business and executives. Click here for more details on business intelligence and data mining.

Now you can make strategic decisions on the fly, without the need for time consuming data manipulation, without lengthy data transformation or reorganization and without programming, thus gaining precious time and achieving major savings.

Applicability and Range of Decisions

Auguri's platform is designed to handle all types of decisions from simple decisions such as the selection of a laptop to complex decisions such as the launch of the Space Shuttle. The difference between a simple and a complex decision is typically the number of criteria involved and the complexity of their behavior. For example launching the space shuttle is a complex decision because it involves many criteria such as the O-Ring, the Fuselage, etc. It is not an easy task to define the behavior of these criteria and their dependency on factors like Temperature, atmospheric pressure etc... Some of the key applications where Auguri's technology has been used in the field of Decision Support include:

Threat analysis

a. Enterprise Search
b. Selection (Resource allocation)
c. Triage
d. Decision Support
e. Business Intelligence
f. Intelligence
g. Prioritization
h. Procurement
i. Root Cause Analysis
j. Threat Analysis and Risk Assessment

By having a unique platform to make all types of decisions in an organization, you reduce the need for training. There is no need to train your staff on a variety of software to handle the different decisions they face.

Products

All Auguri's products are web based. They run off a server and can be accessed through a web browser.

The architecture deploys the solution as a web service. It is comprised of the following components:

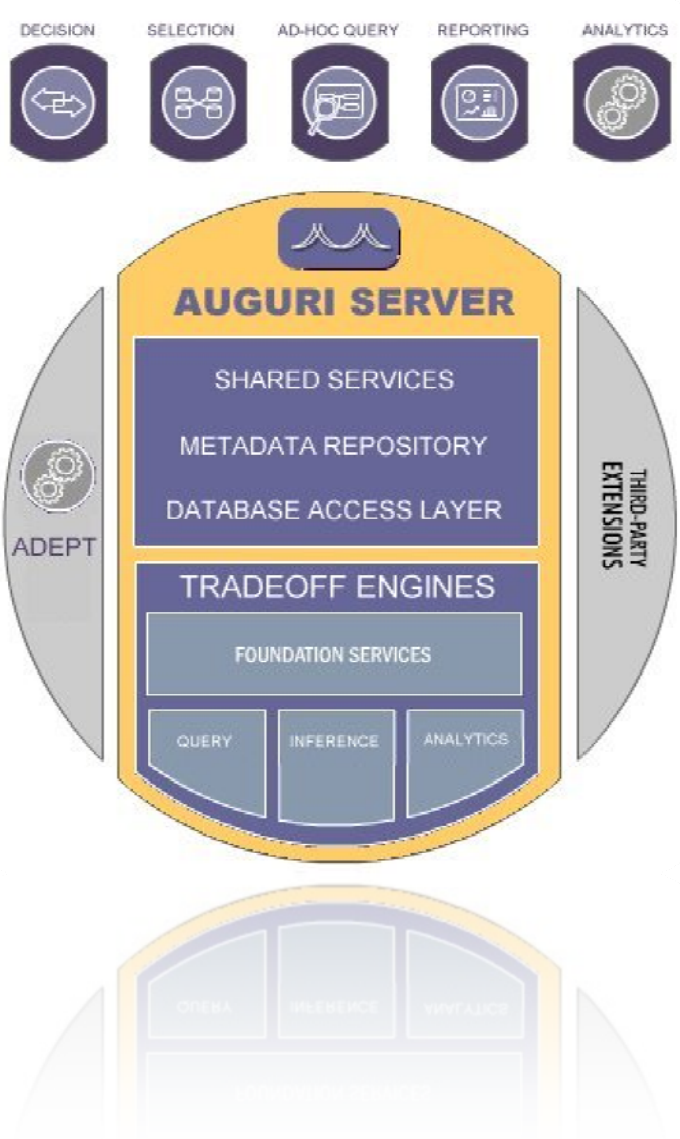

Figure 4: Auguri Data Server and product suite

- **Auguri Server** encapsulates the functionality of Auguri's 3 engines (i) the tradeoff based query engine, (ii) the inference engine, and (iii) the analytics engine.
- **ADEPT** is Auguri development platform. It builds applications that leverage Auguri's server functionality. These developer tools enable the rapid creation, management and deployment of tailored enterprise decision support applications - with or without programming knowledge- that can grow and change with customer needs.

  The Auguri server may be also accessed through a programmatic interface (APIs , DLLs)

The current release of Auguri is 4.1. The Data Server is robust, solid and proven as it has been licensed and used for about 8 years. The first release of Auguri dates back to 2000. We continuously improve its performance and innovate its functionality. Auguri has been licensed to large organizations such as the Boeing, the United Nations and the Defense Acquisition University to list a few.

Auguri Creativity Suite includes an annual license of the Data Server for development purposes only, and the development tools, along with some basic training, and technical support is priced at $60K. We have found this Suite to be very valuable at conveying the power of our innovation. For example, after licensing the Auguri Creativity Suite 3+ years ago, Boeing is today at its 5th contract with Auguri.

The production license price depends on several factors. It is typically priced at $100/MHz of Server CPU (which is a reflection of the amount of usage of the software). A Server running on a CPU of 2GHz can serve for an average application 50 concurrent users without performance degradation. This configuration is licensed at $200K.

Competitive Landscape

As a customer articulated it: “Auguri puts the *decision* in decision support systems”

At a high level the key players in this field as identified by Gartner include Cognos, Informatica, Oracle, Microsoft, SAS, Business Objects and the likes. Most of these solutions are focused on the reporting aspect and the data summary of business intelligence. As a result Auguri's solution tends to complement these solutions by focusing on the decision process itself.

On the other hand there are a couple of alternative DSS metaphors in the market that take a radically different approach to addressing the decision challenges. Constraint Based DSS select a solution by a process of elimination such as collaborative filtering. The main drawbacks of this techniques is that it is time consuming, requires some level of expertise and has a main technology limitation: criteria are handled in a serial manner. Rule Based system include AI, Decision Trees, Predictive Modeling, Influence Diagrams or Bayesian approach or a combination of the above. These techniques, while occasionally powerful, lack flexibility, require significant programming and tend not to deliver very relevant results in a dynamic environment. Auguri's DSS offers a leap over traditional techniques such as rule based or constraint based systems:

	Rule Based	Constraint Based	**Auguri**
Decisions:			
Tradeoff Enabled	NO	NO	**YES**
Customized per user	NO	NO	**YES**
Used to Justify and Understand	NO	NO	**YES**
Handles Uncertainty	NO	NO	**YES**
Flexible	NO	NO	**YES**
Predictable	**NOT ALWAYS**	**YES**	**YES**
Based on need assessment	NO	NO	**YES**
Performance:			
Relevance of Results	NO	NO	**YES**
Total Time to Final Result	Lengthy	Lengthy	**Rapid**
Data Manipulation Needed	Some	Extensive	**Minimal**
Coding Required	Extensive	Average	**Minimal**
Development Time	Average	Lengthy	**Rapid**
System:			
Easy to use	NO	NO	**YES**
Practical for novices	NO	NO	**YES**
Share Expertise Know-how Raise Level of Novices	NO	NO	**YES**
Saves Expert's Know-how	NO	NO	**YES**
Wide Range of apps - Reduces Training Needs	NO	NO	**YES**
Low TCO (build, impelem. & maint.)	NO	NO	**YES**

EXHIBIT A: Underlying Technology

Challenges of Parametric Search

Parametric search represents the most common approach to today's querying needs, a fact which is demonstrated by the pervasiveness of SQL. Under certain circumstances and provided that query parameters are precisely crafted, this model can be quite effective. In general, however, parametric search suffers from two very significant problems.[1]

Firstly, it often fails to return an appropriate number of results, instead returning far too many or too few. Being able to control the approximate size of a result set is important. A query that generates a list of e-mail recipients for a marketing campaign is ineffective if it yields only three results; a search which yields one thousand results for personal computers is equally ineffective. Inappropriately sized result sets force users to run additional queries, broadening or narrowing the search parameters. But without an intimate knowledge of the data set, crafting a query which is appropriately precise may take many iterations. Thus, poorly sized result sets are an inefficiency in time and effort.

Secondly, SQL queries often fail to deliver the most relevant results. In particular, such queries cannot distinguish between search criteria with varying degrees of importance. As an example, suppose we are searching for inexpensive hotels which are as close as possible to San Francisco International Airport. Shouldn't the results be differently prioritized for users who are price-sensitive, as opposed to users who are concerned with proximity to the airport? Parametric searches cannot make this distinction.

How Parametric Search works

A parametric search imposes explicit search constraints on one or more parameters in the dataset, extracting only the data that meet those constraints.

SELECT ATTRIBUTE FROM DATASET WHERE CONDITION 1 AND CONDITION 2 etc..

Constraint-based search can be represented in n-dimensional space, where *n* is the number of attributes, by a box based on the search constraints. The search retrieves the data that meets the query constraints – that is, the data inside the "constraint box".

Explaining SQL Limitations With Optimization Theory

Optimization theory teaches us that ideal results are usually found close to the pareto-optimal boundaries – these boundaries are typically the intersections of curves, planes and graphs representing constraints and utility functions. In the case of parametric search, this intersection corresponds to a corner of the "constraint box". The optimal results in a parametric search will land next to the corner of this box which represents the ideal result. This observation leads to an important realization: **Parametric searches often miss highly relevant results** that are just outside the search box (but are close to the optimal corner), despite their attractiveness to the user. This occasionally leads the DoD decision makers to make poor decisions.

[1] The terms *parametric search*, *constraint based search*, and *SQL-type search* are synonymous and will be used interchangeably throughout this paper.

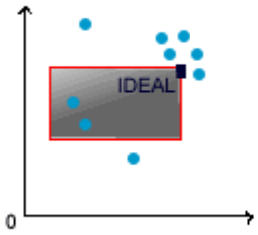

Figure 5: Ideal matches in SQL queries reside as near a corner of the search space.

In the example above, there are a number of data records very close to the ideal, but because they reside outside the query boundaries, they will not be found by the parametric search.

Paradigm Shift

We have also seen that relevant records that are only slightly on the wrong side of a query boundary will be ignored by parametric searches. An alternative solution to this dilemma would be to rank all records based on proximity to an ideal point. This would ensure that no relevant records would be missed simply because they were outside the search constraints. Furthermore, it would ensure that the right number of results could always be returned, since this number could now be set explicitly.

Enriching Search with Tradeoffs

Having solved the problem of excluding relevant search results, we are positioned to address the issue of relevance. Introducing the concept of tradeoffs – the relative importance of various search criteria – provides a very elegant solution. Tradeoffs can be captured by a set of weights corresponding to each criterion. These weights result in a stretching or compressing of the corresponding axis. For example, if an Auguri user defines price as extremely important relative to performance, the axis corresponding to price will be stretched. This will effectively magnify price differences to make them more significant – data records which are more expensive will appear *even farther* from the ideal.

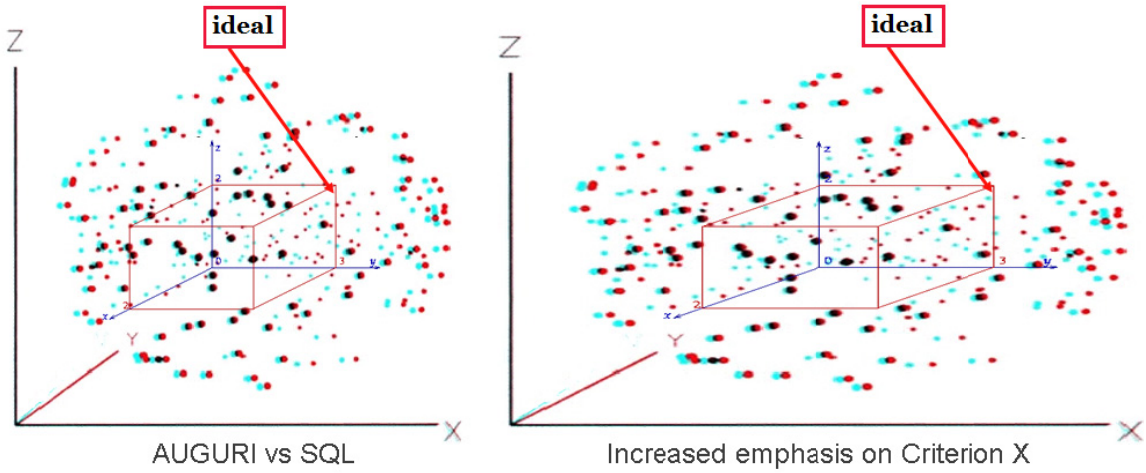

Figure 6: Parametric Search vs. Tradeoff Search

This technique allows us to generate a mathematical notion of record "score", defined as the geometric distance from a user ideal, which is useful. The score of a record can be expressed:

$$\sqrt{\sum_{n=1}^{N} w_n (x_n - x_{in})^2}$$

where N is the number of criteria, x_{in} is the nth coordinate of the ideal, x_n is the nth coordinate of the record, and w_n is the weight of the nth criterion.

By adhering to a new model for querying which is *tradeoff-based*, Auguri searches can effectively emulate the way humans think. As an example, a tradeoff-based search model would allow commanders to rank search results in accordance to how closely they match the ideal situation. This is a key advantage, because in war not only do selection criteria change rapidly, they conflict greatly, and very seldom will a totally ideal situation exist. Selection criteria are traded off, and results should be ranked according to how closely they match the ideal solution.

In different areas of Iraq for example, criteria may be greatly different. In the area supportive of terrorist insurgents, are troops available who are fluent in Arabic? Are local troops adequately trained, supplied and manned? Tradeoff based search provides ability to intelligently weigh various criteria, and prioritize possible courses of action.

Inference

Tradeoff based approach enabled Auguri to develop a particularly powerful tool: The ability to reverse engineer the query. The idea is that given a result set, the relative weights corresponding to each criteria can be derived. Put another way, given a list (partial or complete) of "best" results, it is possible to infer which criteria are most important.

Solving the inference challenge requires calculating the weights of each criterion in a set of inequalities where D is the score of a given alternative and M is the number of ranked results:

$$D_1 < D_2$$

$$D_2 < D_3$$

...

$$D_m < D_{m+1} \rightarrow \sqrt{\sum_{n=1}^{N} w_n (x_{mn} - x_{in})^2} < \sqrt{\sum_{n=1}^{N} w_n (x_{(m+1)n} - x_{in})^2}$$

...

$$D_{M-1} < D_M$$

This system of inequalities can be used to derive a system of weights compatible with such an ordering. We normalize the weights by having $\sum w_i = 1$, and we optimize the result by selecting the weight array that causes the least disturbance from the weights obtained from the previous session; $\lim \Delta w \rightarrow 0$.

The ability to infer a query from a result is an incredibly powerful tool which opens the door to a whole suite of capabilities. Inference can provide real time intelligence at a click of a button. Now, vast amounts of stored data can be used for analysis and true decision support.

Exhibit B: Authors

Fadi Victor Micaelian

CEO – Auguri Corporation
EDUCATION: MS in Engineering from MIT and an MBA from INSEAD
CURRENT POSITION AND RESEARCH: CEO and Founder of Auguri Corporation
Mr. Micaelian history in research dates back to 1986 when as a graduate student at MIT he was a Research assistant for a project on “detection of underground fluids”. His strength lies in a solid foundation on mathematics combined with his love of physics and his years of experience in enterprise software. His has spent years focusing on R&D in the field of decision support, decision simulation, selection, tradeoff based theory and search technology.
RELEVANT EXPERIENCE: Mr. Micaelian is the CEO and Founder of Auguri Corporation where he brings over 20 years of experience in the software industry. Prior to founding Auguri, Mr. Micaelian worked at Oracle Corporation from 1987 to 2000 and held various positions in product management, marketing, business development, strategic alliances and engineering management. Mr. Micaelian is an inventor and was awarded several patents by the USPTO. He has been a keynote speaker at multiple international conferences and has authored several papers in high technology, decision support and innovations. Mr. Micaelian holds an MS in Engineering from MIT and an MBA from INSEAD

Emil Scoffone

CTO – Auguri Corporation
EDUCATION: PhD from UC Berkeley
CURRENT POSITION AND RESEARCH: CTO Auguri Corporation
Mr. Scoffone PhD research was in NMR spectroscopy, with an emphasis on designing new experimental techniques and the processing, presentation, and analysis of large spectral data sets. He did research work on developing metaphors for object-oriented database access (at Oracle); on extending the Adobe PostScript kernel to support specialized imaging algorithms and input devices like scanners (at Adobe); on the architecture of large-scale distributed applications (Carnelian); on developing models for efficient product catalog management and product recommendations (at On-Link); and, of course, on the development of tradeoff-driven decision making models and algorithms (at Auguri).
RELEVANT EXPERIENCE: Mr. Scoffone brings to Auguri over 15 years of experience in research and development. Before joining Auguri, Mr. Scoffone was a Principal Architect at OnLink Technologies / Siebel Systems, an Internet startup acquired by Siebel in September 2000. Playing a significant role in the engineering of the early versions of their products, Mr. Scoffone held development and architectural positions at Oracle Corporation, Adobe Systems, and Netscape. He holds a PhD from U.C.Berkeley

.