

Personalization, Learning and Guidance Technologies

TRACK DESCRIPTION

WatchIT.com™ brings together the best in IT. Our monthly programs feature technology experts, business leaders, and industry analysts who offer a comprehensive perspective on the trends, issues and technologies important to today's IT professionals. Understanding, advice, best practices: you get all of this and more from watching WatchIT.com.

This program is part of our Core Technologies track, which focuses on technical insight and the underlying concepts of essential technologies.

INTRODUCTION

You know, today's IT managers are walking a tightrope.

On the one hand, our end users have very high expectations for personalization of both Web content and their customer service applications.

On the other hand, us folks in IT are given the mandate to automate as much of this as possible.

How do we deal with these juxtapositions, where on the one hand we have high expectations for personalization and on the other hand we have to do IT automation.

Our discussion today is going to take a look at the current state of customer service technology. We're going to look at:

- ~ Telephony systems;
- ~ Routing systems; and
- ~ Web-based personalization systems.

But more importantly, we're going to look at the technology behind these systems.

- ~ We're going to understand voice recognition systems and how they can be used in both telephony and Web-based systems.

~ We're also going to take a look at a very exciting technology – concept-based search and retrieval systems – that promises to give our customers the information that they need very quickly.

~ We're also going to look at what's happening today in Web personalization and data mining technology, to understand how we can track and correlate our end users' behavior, in order to give them the level of personalization that they think they deserve.

Hi, I'm Don Burleson and welcome to WatchIT.com's Core Technologies track, Personalization, Learning and Guidance Technologies.

I have a background both in computer science and business administration. I have an MBA and I've worked for 20 years in the full-time design of database architectures, database designs and database administration.

I'm the author of 10 popular database books and I also serve as Editor in Chief of Oracle Internals magazine. I'm also a regular contributor to magazines, but I have a lot of industry experience to tell you about today as well.

I've consulted with numerous Fortune 50 companies, working with data warehouse projects, as well as customer service applications for numerous dot coms.

I'm here to share my experience with you on how you can leverage your customer service applications in order to get ahead of the curve.

AGENDA

Today we're going to be taking a look at several things.

~ We're going to start with an overview of the existing customer service technologies, routing technologies, and Web personalization technologies.

~ We're then going to take a look at the shortcomings of each and understand how the misapplication of this technology can often lead to disastrous results from an IT perspective.

Then we get into the fun part. We'll take a look at a technical discussion:

~ We'll rip into the bowels of voice recognition systems and understand, at the conceptual level, how they work.

~ We'll take apart concept-based search and retrieval engines – understand how they parse human English and retrieve information from very large databases with sub-second response time.

~ And most exciting of all, we'll take a look at Web personalization technologies – how we can customize the content of our Web page and use the rich data mining techniques available today, in order to bring our customers closer to the information that they want and desire.

ROI

After completing the session, you'll have a very general understanding of the concepts behind the latest customer service technology, and you'll understand the application of these technologies within your own department.

You'll also appreciate the IT challenges involved in the acquisition and implementation of customer service technologies, including voice recognition and Web personalization.

AN OVERVIEW OF EXISTING CUSTOMER SERVICE, ROUTING, AND WEB TECHNOLOGIES

Let's start with an overview of customer service technologies, and take a look at how the technology is changing.

One of the problems, with all telephony systems, is the limitation in responses that the customer can make using their numeric keypad on the telephone as an input device.

We're also starting to see the same sort of limitations on Web-based applications, where people would much rather speak or in some other fashion communicate their desires, rather than typing out 14 digit product numbers or trying to convert letters to numbers on a telephone keypad when using telephony types of systems.

The Psychodynamics of Customer Service Systems: The ELIZA Experiment

To understand how this works, let's take a brief exploration into the psychodynamics of these customer service systems.

I'm reminded of a system that was created in the 1960s as part of a psychology experiment at MIT. The computer program they used was called ELIZA.

They took psychology subjects into a large room full of blinking lights – a very impressive MIT environment – and told them that they were going to sit down at a keyboard and converse with the world's most intelligent supercomputer.

In reality, Eliza was a very simple software program that had been pre-programmed to anticipate the subjects' responses in order to appear intelligent.

One of the things the researchers at MIT had discovered, for example, was that it was not uncommon for a person to ask, "How are you Eliza," to which Eliza would reply, "We're here to talk about you, not me."

These kinds of things gave the subjects the impression that the customer service, in this case Eliza application, was far more intelligent – it was truly listening to their needs.

Interestingly enough, the MIT researchers also discovered it wasn't uncommon for the subjects to type in profanity words, to which Eliza would simply respond, "my, my, I'm not accustomed to such language."

The point of the Eliza experiment is that we in IT can design customer service input devices such that we actually give the customer the impression that they're getting personalized service, while at the same time taking advantage of the automation. And that's what we're really all about.

Fax Distribution Technology

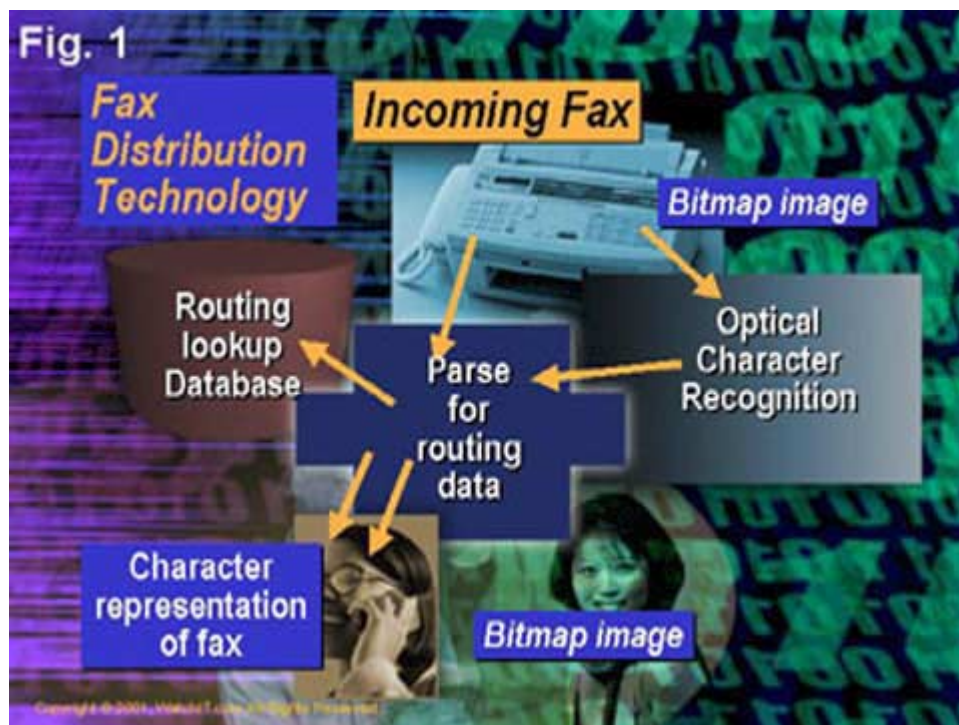
If you want to understand customer technology let's begin with a simple discussion of fax distribution technology. Fax distribution technology is a good starting point for our discussion today because it doesn't really have any direct human interaction.

In a fax distribution network, incoming faxes are received, parsed, interpreted and then routed to the appropriate department within the corporation.

Internally, of course, the fax is received as a bitmapped image. But then the bitmapped image goes through optical character recognition (OCR) where it's turned into word tokens. These word tokens are in turn looked up in a routing look-up database, and the entire fax bitmap image and the character representation of the image is sent to the appropriate department.

Again here, a perfect example of using IT technology to automate something that is routine, mundane, and the kind of thing that is perfectly appropriate.

Figure 1



Automating Telephone and Web-Based Customer Service Front Ends

We get into a little more of a challenge when we start talking about automating customer service technologies using telephone and Web types of front ends. Today's customers expect personalized care. If you go out to Amazon, you'll notice the personalized content pages that they deliver.

People expect the same level of service on the telephone and on the Web that they might get from a brick and mortar establishment. And the question is how we can create telephony applications to service these kinds of customer needs. The frustration with telephony applications is evident to anyone who's ever had to sit through 10 minutes of listening to "press one for this" and "press two for that."

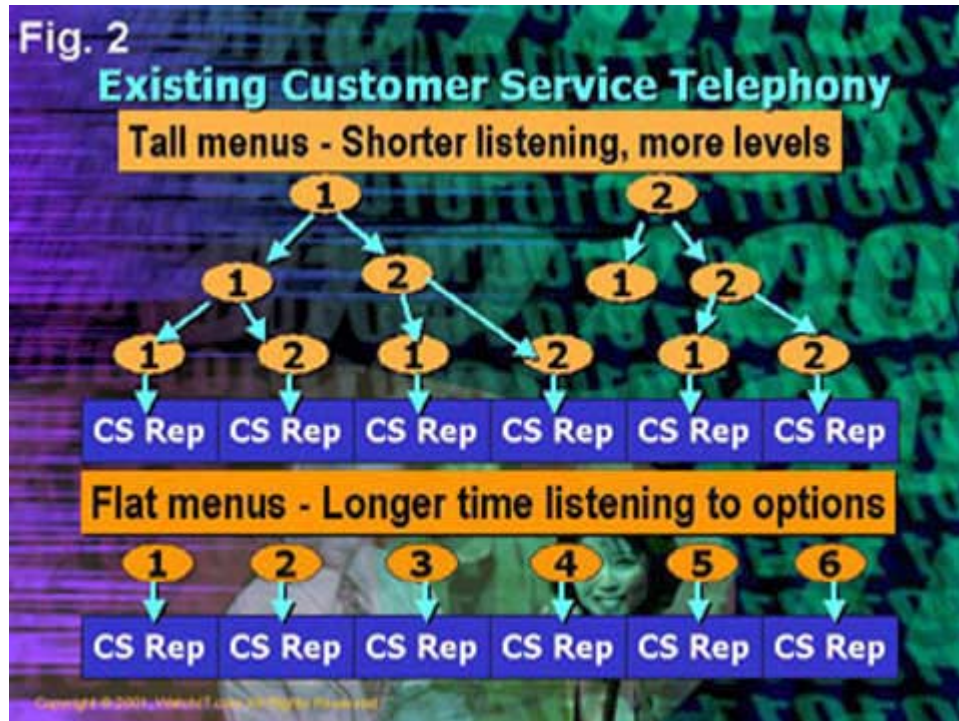
The problem is the routing technology has been poorly designed by IT management. As we saw in our example of fax technology, the technology works very well – it's up to us to understand where we must introduce human experts into the telephony types of systems.

Telephony Systems: Tall Menus and Flat Menus

When exploring a telephony system for customer service, we generally find two categories of menus. We can find tall menus and what we call flat menus. Flat menus were originally

designed in order to get the customer directly to the customer service representative quickly by only having them make one or two responses.

Figure 2



The downside, of course, to the flat menu is that they have to listen to “choose one for this” all the way through “choose 9 for that” before they can actually make a response and drop down. People who use flat menu systems uniformly agreed that they didn’t think that they were ever going to actually get to speak to a real human being.

The ones that have proven more successful are the ones that use taller menus, where each cascade through the menu is very short, concise and to the point – “press one for customer service, press two for order processing, press one for accounts receivable, press two for finance” – such that the end user is quickly dropped through and routed to the appropriate queues within the customer service representatives.

But the technology doesn’t end there. Customer service technology is far more than just routing the calls to the appropriate departments. Within a large customer service organization, we might have 10 or 15 departments, each with a dozen or more customer service representatives manning the phones. The customer service telephony application has to do its own load balancing.

Case Study: Microsoft’s Telephone Support System

If you want to take a look at a good system that has worked very well, we can take a look at the telephone support system for Microsoft Corporation. Microsoft is very indicative of the types of

customer telephone systems that have dozens of departments, each with many dozens of people working the telephones. Their telephony applications do far more than routing.

Along each step of the way, the human touch is interjected, and at the same time, the software does queue reporting and load balancing to ensure that the customer gets placed with the highest probability of seeing a human as quickly as possible. Let's take a look at how this technology works.

Basically the customer comes through, and through a series of cascading menus specifies the department that they need technical support from. At that point, they're put into a response queue, and they wait their turn along with everyone else for the next available person within that department.

Figure 3



But it doesn't end there. The load balancing software, as we talked about before, automatically knows the average wait time within each queue and also knows the rate at which each customer service representative is servicing the client.

If an average call in the FoxPro support queue is 2.5 minutes, they can compute that and based on typical mathematics, route the calls to the different people within the FoxPro queue who can give the customer the fastest service.

Herein is the genius behind the Microsoft system:

They essentially hired a human being to monitor this system, which may be having hundreds if not thousands of telephone calls at any given time and this person is what Microsoft calls the DJ.

Instead of having the end user listen to elevator music, classical music, or something calming because they have no idea whether they'll be waiting five minutes or five hours or if ever a customer service person is going to answer, the DJ gives them an exact idea of how long the wait times are within each of the queues.

Again, a very inexpensive way of getting a lot done, in terms of customer satisfaction.

The customer may still have to wait 20 minutes in order to get to the Microsoft C technical support person, but they know coming in that they have a 20 minute wait, and by listening to the DJ they can gauge how long they're going to have to wait their turn in order to get customer service.

Here is an example of a perfect marriage between telephony applications, taking care of the routing and load balancing, and interjecting the human touch on a very massive scale, such that only one human being can manage the routing of thousands of telephone calls per hour with a high degree of success and high reports of satisfaction from the end user community.

The Shortcomings of Customer Service Learning and Guidance Technologies

Now that we've reviewed an example of a good customer service technology, let's take a look at the shortcomings of the customer service learning and guidance technologies.

Remember, it's not the technology itself that our customers are complaining about – it's the short sighted implementation of the technology.

We see, from samples of the customer community, that they have a very deep frustration. They feel depersonalized, they feel that they're not getting to human experts soon enough.

Even more, we see the reflection of our own IT designs mentioned in the customers' complaints.

They tell about feeling lost within the telephony application, or even within a Web application, where screens link randomly from one to another and the customer doesn't really understand where they are in the application itself.

An even more pressing concern from the end users has to do with Web personalization, which we'll be talking about later on.

The big privacy concerns – you'll see television commercials where people want their anonymity when viewing the Web. But at the same time, it's our job within IT to watch what they're doing, to play big brother, to use unobtrusive measures to watch what the people are doing on our Web site so that we can service their needs better. We need to target the marketing towards the people based upon their existing patterns of use and purchases within our Web-based application.

Again, these privacy concerns are very compelling because the end users don't want people looking over their shoulders, while at the same time, us folks in IT are mandated to look over the shoulders of our customers in order to better serve their needs.

Measuring Human Costs vs. Software Costs

Let's take a look at the costs of these kinds of systems. Basically we're talking about measuring human versus software costs.

It's very easy to imagine abandoning the Web entirely – simply go back 10 or 15 years ago when everything was dealt with by human beings.

Part of the problem, of course, with today's corporations is that the customer service technologies are too broad.

We may have dozens of systems, dozens of areas for which customers will call in, and the problem is finding an empowered human being who is capable of solving the problem of the person who is calling.

To find a corporation that can empower people at that broad a scope is impossible. Hence, we must have some kind of routing technologies involved in these customer service applications. The question is where to introduce the human touch to these customer service applications.

It's very easy to just have someone go through a series of automated menus. It's kind of like the mailbox system where people are sorting the mail, putting it in all of the different bins, when behind the scenes it's all going in to one big bucket.

These are the kinds of feelings that customers sometimes have when cascading through endless series of menus, either voice menus on the telephone or menus on a Web-based system.

We have to be able to interject humanity or intelligence into that system in order to ensure the customer that they're going along the right route, that they can trust the system that they're actually using.

Measuring Effective Customer Support

When we talk about the IT issues involved in this, we have to talk about measuring effective customer support. What are the metrics we use for measuring effective customer support?

Basically, these types of telephony systems and Web-based systems can measure the total elapsed time from the time the person entered the system until their call is closed. We can also take a look at the time they spend navigating the menu structures, the time they spend in each one of the response queues, and the time they spend talking one-on-one with the customer service representative.

Now, while these metrics are useful, they don't really give us an end gauge of customer satisfaction unless we actually tie that to repeat calls, because if customers need useful information and they don't get it the first time, they may come back to our system again and attempt to reach us.

So we have to remember that sometimes measuring the effectiveness of these customer service systems means going out of the box. We have to actually get in touch with the end users.

For example, look at customer support companies such as Hewlett-Packard. I called Hewlett-Packard for tech support about a month ago, due to a problem with an HP server.

I went through the cascading menus, I reached a highly-qualified technician who answered my questions and the next day I received a follow-up call from a real human being asking me what I thought of their customer service system.

This is an example of the type of system that is committed to the effective use of the technology.

But that brings us as information technology people into involvement in the cost and staffing issues of these kinds of systems.

TECHNICAL DISCUSSION: VOICE RECOGNITION

When we take a look at the technology behind these customer service applications, we have to take a look at it from an IT perspective.

It's one thing for our end users to come to us and say, replace the numeric keypad on our telephony application with a voice response system.

Let's rip into this technology and take a look at how it works.

Speech-Enabled Interactive Voice Response Systems Tutorial

Essentially, voice recognition technology deals with pattern matching. That's why you will find that the implications of these systems tend to be very large in terms of RAM memory consumption.

Most of the ones that are used within commercial applications – like telephone directory assistance – are generally processed on large UNIX servers where the processing power is not as important as the amount of RAM memory.

These systems generally have between two and four gig of RAM which is fully populated with pattern matching arrays, so that these systems are able to deal with what they call generic speech files.

Rather than a system, for example, that does voice recognition on a PC, which gets to learn your own voice inflections and pronunciations, these types of global systems have to be able to deal with any kind of a dialect, from a Texas drawl, a southern twang, to a cockney accent.

And it has to be able to quickly and accurately parse through and determine the character representation of those spoken words.

We also have to deal with gender differences – tonal differences between males and females. We have to deal with pronunciation issues between guttural languages, such as, German and other languages, which slang together, such as French.

All of these things are taken into consideration when building these global types of networks.

Now let's move on and take a look at concept-based search and retrieval, and see how voice recognition technology can be extended to the next level.

TECHNICAL DISCUSSION: CONCEPT-BASED SEARCH & RETRIEVAL

Now that we've talked about how spoken words can be tokenized, or broken down into discreet components, the next important thing is how we can take a look at analyzing those words to derive the meaning.

Once we can derive the meaning from the spoken words, we can direct our customers to the most appropriate information that they're looking for. And let's take a look at how some of these concept-based search technologies work.

It's important, before we get into this, that I note that these technologies are out there today. There are three major vendors who work with these types of concept-based search engines and let's take a look internally at how these things work.

Initially, what these concept-based search engines do is take the written request and parse it. The very first thing they do is they remove noise words – of, a, than, it – and deal with what's left over, the nouns, the adverbs, and so on.

And they essentially take these word forms and modify them. The first thing we see in these engines is stemming. For example, I might have a query about a house, in which case the synonym would be expanded to deal with the stems of the word house, including housing, and home.

Once all of the stemming is done, we go through a synonym expansion phase. The synonym expansion phase is where we actually take house and include other synonyms for it: apartment, dwelling, abode, and so on.

We wind up building very large Boolean queries where each query is "anded" together. But within each token, house, home, abode are "ored" together.

We wind up with these very long strings “anded” together and each token being “ored” together, where I have house or housing or home or abode and so on.

On-Point Queries

A lot of these tools measure their results along two dimensions: precision and recall.

Precision is how well these concept-based search technologies retrieve what they want and recall is the measure of the amount of on-point responses that these tools return to you.

The technologies today actually allow you to highlight a paragraph of text – say a judicial decision – that’s on-point and you just click a button that says “show me more like this.”

I’ve seen cases with wide synonym expansion where you can highlight a case on maritime law, and return cases where the judge talks about two ships that pass in the night – not exactly an on-point kind of query.

This brings me back to the central theme, that again, the problem is not with the technology itself, but with us in IT and our implementation of the technology.

When we start using these concept-based search engines we must very carefully monitor the level at which we do stemming and synonym expansion, so that we can get a high degree of both precision and recall from our concept-based queries.

Language Translation

If we take a look at the early works in parsing human English, we find some very interesting tools. Some of you may have gone out on Yahoo on the Web and used their BabelFish tool.

I often get queries from people who’ve read my books, which have been translated into German, Korean, and they often assume that because I wrote a book that I knew how to read it.

I often get queries in Spanish and I have to actually go in and use language translation tools, translating their Spanish into English, so that I can in turn respond to them, translating my response back into Spanish in order to directly address my own customer base.

Some of these early attempts at language translation were conducted by the Central Intelligence Agency (CIA). Specifically, Professor Liz Liddy at the University of Syracuse, did a lot of work with database queries where you could query a database in German, the database is stored in English, and spit out the results in Russian.

A lot of these early translations worked very well for well-structured kinds of English, but when we start dealing with concept-sensitive grammars and the natural ambiguity of the human language, we start running into problems. A good example is one of these early systems built to translate English into Russian and then back into English.

If you gave it the phrase, “The spirit is willing, but the flesh is weak,” translated it into Russian and then from Russian back into English, it said, “the Vodka is good but the meat is rotten.” Not exactly a literal translation of that.

Addressing the Natural Ambiguities in English

Today (2000) the concept-based engines have been making great strides in dealing with these kinds of natural ambiguities in spoken English. Again, this gets back to the word recognition, where we’re parsing context sensitive grammars of “Please write to Mr. Wright right now.”

When I worked at the University of New Mexico – the Harvard on the Rio Grand, we used to call it – one of the professors in our engineering department formed a company called Excalibur (now Convera), which today, 20 years later, has grown into one of the major concept-based search engines.

Back in the 1980s, this professor came up with a product he called Savvy. And Savvy was supposed to be a spoken interface to a database.

At the time, in the 1980s – this is before relational databases – we didn’t have easy-to-use access languages, such as SQL.

And this professor had tried to come up with a free-form query language that would essentially take a free-form request and translate it into the necessary data manipulation language to pull the interesting data out of the database.

Savvy worked very well with well-structured commands. You might say, “What is Joe’s salary?” and it would come back with the correct number.

But it failed with the natural ambiguity of queries, such as, “How long has Joe been with us?” I remember typing this query into Savvy and it came back and said, “What do you mean ‘with us?’ Are you asking for Joe’s date of birth or Joe’s date of hire?”

These are the natural ambiguities that we have a problem with in these types of concept-based search engines. A lot of phrases that we have also deal with this. For example, you can tell a concept-based search engine “Mary had a little lamb.” The concept-based search engine will not understand how to parse that verb “had.”

Mary had a little lamb – did she eat this lamb?

Mary had a little lamb – did she own this lamb?

Do we take a biblical interpretation of “had?”

What do you mean, “Mary had a little lamb?”

So we still see these problems in concept-based search engines, where they cannot always resolve the natural ambiguity of the English language.

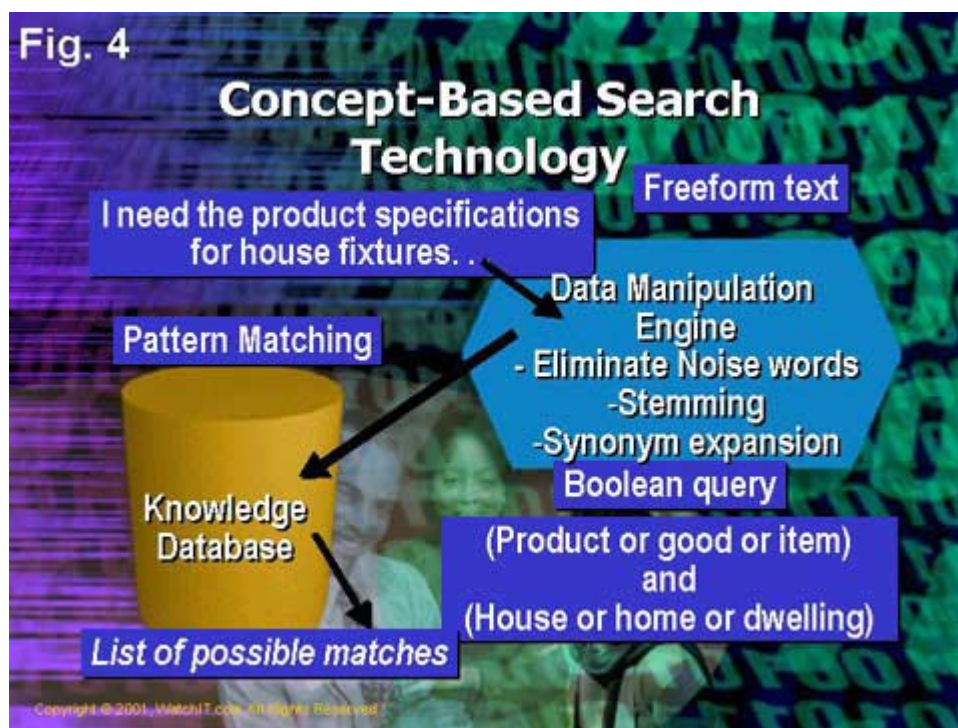
Concept-Based Search & Retrieval: How It Works in the Real World

Now let's take a look at how these things actually work in the real world. And it's important that we note here that this is not science fiction. A lot of systems in the real world today are using concept-based search technologies in order to help their customers get to technical information that they need, far more efficiently than they ever have before.

Generally, the way these engines work is they allow you to either highlight a paragraph in information that you want to "find more like this," or you can actually enter your own types of queries against these.

We again go through the basic steps of eliminating noise words, stemming, synonym expansion, but here's where it gets different: once we've built this gigantic Boolean query of each noun or token "anded" together and all of the synonyms within it "ored" together, we wind up with a huge Boolean query that has to be serviced very quickly by a knowledge database.

Figure 4



A lot of the tools, such as Excalibur – Oracle also has a tool that does this, and there are other vendors as well – essentially employ a pattern matching mechanism, so that we can search vast quantities of information at very high speeds.

The way these work is, instead of storing the database as text, it's a full index inversion, where the text exists as virtual arrays of indexes, such that this technology allows us to feed a Boolean

expression of hundreds or even thousands of words, and it can return a ranking of likely pattern matches with sub-second response time.

I'm working with a dot com system right now where we've implemented one of these technologies and even on a Linux server, four CPUs with less than four gig of RAM, we can get sub-second response time on very large sophisticated queries.

The problem when you're trying to do this kind of concept-based search technology is that you don't always get back one hit.

Essentially, just with the search engines that we see on the Web today, whether we're talking about Google, Lycos, Yahoo, Magellan, or whatever search engine you like to use, we generally find our response sets brought back in some form of ranking.

This internal ranking is the internal pattern matching probability that this hit is on point with your original request.

And again, the real challenge with these technologies is not getting the data out of the system, but getting it out with a very high degree of precision and recall.

Remember – we don't want to search on maritime law and find hits on two ships that pass in the night. Neither do we want to have to become experts in the pattern-matching engine. We need to simply highlight a paragraph of text, show me more like this, so that we can feed that to a system that has applications within customer service technology.

Imagine a corporation such as IBM. IBM produces hundreds of hardware devices, and people call into IBM all the time trying to find out technical specifications on these devices – is this dms chip compatible with this CPU? – the kinds of questions I make all the time in the IT industry when expanding a database server.

If I had a complete concept-based search engine, I could simply feed it a free-form query and it would go through this pulling the information out of a knowledge base and it would allow me to get my technical information without any intervention from human experts.

Especially within the IT industry, people who are knowledgeable about technical specifications are very expensive people to have, and these kinds of people with generally advanced degrees are not willing to work telephone help lines.

So we're starting to see this type of concept-based search technology making inroads even within customer support of organizations that deal with IT hardware and software.

Concept Based Search Engines: Challenges to the IT Organization

Let's talk about the challenges for the IT organization with concept-based search engines.

The biggest one we have is that there are a limited number of vendors. Generally, we see three out there today who are dominating the marketplace. We see:

- ~ Excalibur Corporation that we talked about originating in Albuquerque;
- ~ Oracle interMedia, which offers a product that does concept-based searching; and
- ~ A third company called NEXTpage, which was formerly Folio Corporation out of Salt Lake City, Utah, which does excellent work. Some of you may remember the old Folio Info Bases on concept-based searching and full word inversion types of tools.

But we have to remember that a lot of companies who are seeking a competitive edge in today's marketplace are developing their own concept-based search engines. A lot of dot com's and a lot of companies for which it's very important for the end users get to the information quickly are delivering their own proprietary, secret tools.

Proprietary Solution: Google

A good example of this is the Google technology. If any of you have used Google as a search engine on the Web, you might remember that in about 1999 it overtook a lot of the search engines by virtue of two dimensions:

- ~ It was extremely fast, even on extremely long queries; and
- ~ It was extremely accurate.

It beat out HotBot and Lycos, as well as Magellan, on dimensions of both precision and recall, which shows an example of how concept-based search technology is being used today.

But the real differences are the speed in which it can return the query, and the precision and recall of the returned results. Nothing is more frustrating for an end user consumer than to have to scroll through a bunch of results that are not on point.

The Good News: Architecturally Simple Solutions

Fortunately from an IT perspective, the architecture of these systems is relatively simple. If you go to any one of these vendors, Oracle, Excalibur, or NEXTpage, you'll find that their tools are architecturally separate from the rest of your Web server environment.

They essentially reside on their own server, and the only time they actually communicate with the backend database is when they're actually pulling out the result sets and when they're doing their initial indexing.

As we talked about, a lot of these tools will parse through the text of the individual database, tokenize it into arrays of pointers and use that as the backend for the Boolean search. The only time it ever has to visit the database again is when it actually is going back to pull result sets.

You'll find that many of these concept-based search engines not only retrieve a list of hits, but they'll retrieve the paragraph that they think is on point, highlighting those pieces of the paragraph that are the most interesting to the person who is doing the query.

So from an IT perspective, there are vendors who service these kinds of tools.

Computer scientists with advanced degrees can be hired to write these kinds of tools.

There are freeware components to these tools, such as the Princeton WordNet, which can be used to build your own synonym expansion techniques.

The Need for Semantic Experts

But we also see that we have a need within the IT community for someone who is constantly monitoring and controlling these parameters within the concept-based search engine system.

We talked about the need for highly-specialized jargon in some systems. It's actually brought out a new type of data administration role within IT where we have semantic experts – people who are charged with actually controlling the level of stemming and synonym expansion within our concept-based engines and how well our concept-based engines function in helping our customers get to the information that they need.

TECHNICAL DISCUSSION: WEB PERSONALIZATION AND DATA MINING

As we talked about, one of our goals in customer service technology is to be able to track our end user's behavior across the system.

This means essentially tracking the end user, whether anonymous or non-anonymous across our use of the Web domain and keep that information for later use for analytical classification.

This type of technology is a direct outstretch of data warehousing technology.

A lot of these systems begin when the end user comes into the system and we retrieve a cookie off of their personal PC.

We immediately know where they are, in which case we can track them with a very high degree of precision, as they visit all of the different Web pages.

If an end user comes to our Web site anonymously, the best we can do is track the first three octets of their IP address, unless they have a dedicated IP, in which case we will only generally know which IP server they're coming from, and we don't know which one of the thousands of those people they are.

So, essentially, our tracking is done along two dimensions:

~ We have to be able to track those people who are anonymous to our Web site; as well as

~ Those people whom we can quickly identify with the use of cookies.

The History of Data Warehousing

We have to remember that this is an outgrowth of technology that's been with us for decades. Data warehousing has been with us since the very earliest days of data processing.

Even back into the 1970s, I can remember doing data warehouse technologies on tape, where I would have to spin through 10 or 15 tapes doing analytical remunerations. I would have response time of approximately a day reading through all of these tapes. This was the very same kind of thing that today's technology can do far faster, but the goal was exactly the same.

Interestingly enough, data warehousing was essentially invented by a gentleman named Bill Inman.

Bill Inman was a fellow graduate from the University of New Mexico, Harvard on the Rio Grand, and Bill actually owns the copyright to the term "data warehousing" and is considered the father of data warehousing technology today.

So the data warehouse technology can actually be thought of as a very mature technology.

High-Volume Data Tracking

When we take a look at database servers such as INFORMIX-SE and Oracle 8i, we see that these types of database engines are built for dealing with these kinds of high volumes of information.

The question is: What we do with these extremely high volumes of information that are coming across our database server?

Consider, for example, the traffic on a server such as eBay.

These kinds of systems support hundreds of transactions per second and thousands of page views per minute.

So when we have thousands of page views per minute, how do we go about tracking this kind of information when it's coming in through us at very high rates of speed?

We know from taking a look at the telecom industry that telephone switching systems are capable of dealing with thousands of transactions per second and making sense of that information in order to feed their own billing systems.

We need to take this type of telephone system switching network technology and apply it to what is called data mining technology.

What we wind up with if we build the tracking mechanism – which, by the way, is the easiest part of all of this – is a 10 or 20 gigabyte flat file at the end of each day, as each one of our Web servers tracked each and every one of our end users' page visits.

We're now left with a very high volume of flat file information that we need to be able to make sense about.

Data Mining Technology

Generally, when we talk about this we're talking about the data mining area of data warehouse technology, which, unfortunately for us in the IT industry, is one of the most advanced areas.

Generally, when most companies build a data warehouse they work up from a data mark and begin with simple decision support kinds of queries, simple correlations and classifications.

When we start talking about data mining, the rules change entirely. Here we start talking about profiling, determining unseen correlations between data, putting bots together that go out and read this information, looking for statistically significant correlations between things.

I've done a lot of work in data mining technology, and as you in the IT industry know, there are many vendors out there who offer data mining tools.

But most of these are highly statistical in nature and we often have to deal with an issue I tend to call unusable correlations.

I've worked in point-of-sale data warehouse systems that will return such stunning insights as "long underwear doesn't sell very well in Hawaii," and other such not-so-interesting correlations.

And these are the same kinds of things that we have to deal with in Web personalization technology.

Basically, the way we get our hands around the formidable task of reading through all of this Web tracking information is along two dimensions.

We first have to do profiling and classification, both of the types of products we offer at our Web site and the types of users who look at the information at our Web site.

I've worked in a lot of data warehouse systems where they give them cutesy little names like "DINKS" for "double income no kids" and "yuppies" for people with high salary classifications. These classifications are actually built in hierarchies where we can take a look at people who are looking at specific classes of products within our system.

But once we've developed these classifications it becomes far easier to sort through these reams of millions of Web visits and start to make sense of it.

If you want to see an example of this technology in action simply go to Amazon and look up my books. You'll see under there it says, "people who bought books by Don Burleson also bought books by these authors" – Eyal Aronoff and some of my archenemies in the Oracle industry.

You'll find that these kinds of profiling and classifications were done through Amazon's use of this type of data warehouse technology.

Targeting and Personalization

By profiling classifications of users and correlating those with classifications of products, they can target their marketing far more efficiently than they ever have been able to before.

If you take a look at large corporations, major corporations commonly spend hundreds of millions of dollars a year on mass-marketing advertising, television ads, radio ads, as well as, Web banners and things that we find in the Internet.

These kinds of things can be much more targeted towards people that have a specific propensity to buy the product that we're looking for.

I mean it would have been ludicrous for Amazon to offer an Oracle tuning book to my daughter, who was interested in horse models at this stage of her life, just to give you an example.

By targeting the advertising, we spend our advertising budget far more effectively and get the information out as well as e-coupons.

We see a lot of work being done in the data mining, where e-coupons are targeted to specific classes of users who have demonstrated a propensity to buy a particular area of product. These kinds of tools are also used for predictive modeling, as well as personalization of Web content.

A lot of Web news media services will actually track your behavior, take a look at what you find interesting and then personalize the content of the Web pages.

For example, when I read the New York Times I never look at the sports section. So I would expect to see these kinds of things eliminate the sports section from what I'm looking at and concentrate on those areas like the funnies, which I tend to look at first.

And we're seeing these kinds of personalization of Web content becoming more and more available out there, often to the unexpected dismay of our customers. Why would you expect our customers to be dismayed if we try to follow their every move in order to better track what they're doing and give them better service.

It deals with the whole idea of anonymity on the Web. I've seen a lot of television commercials offering payment methods through credit cards, where they guarantee total anonymity on the Web so no one can see what you're looking at. People tend to have a real fear of being tracked everywhere they happen to go on the Internet.

When we talk about Web personalization technologies we have to remember that this is not a nascent technology. Web personalization, and the data mining and the data warehouse things

behind it, are nearly 10 years old. The only thing that's changed is the high volume of this information and the fact that it has to be delivered to us in real time on the Web.

From an IT perspective, it's very easy to go out and build data mining things that will categorize individual categories of products and correlate them with individual categories of people.

The Challenges Involved in Web Personalization

The challenge to us in IT is to be able to dynamically present Web content based upon that information. We can target someone as being a highly-educated professional who has an interest in a certain type of book, but it's another thing to personalize the content based upon what we perceive that individual to be.

This generally gets into the area of dynamically generating Web pages over the Internet. And again here, from an IT perspective, we need to hire specialized experts in order to get this kind of technology. This technology is not cheap. The data warehousing technology, for example, with Oracle, the world's leading relational database, comes free, a lot of it, with the base package in Oracle 8i Enterprise Edition.

The challenge of course is buying many terabytes of disk storage to keep track of all it, and purchasing the human resources necessary to collect and make sense of all of this information.

To accurately do Web personalization, you need:

- ~ Highly-skilled statisticians;
- ~ You need highly-skilled database administrators who are good at managing these high contents of data.
- ~ You need people who are Web experts;
- ~ People who understand Apache technology;
- ~ People who understand Perl;
- ~ C++; as well as;
- ~ The skills necessary with ASP and other tools to dynamically generate Web content - based upon our own perceptions of individual's buying and viewing patterns.

CONCLUSION

Well we've looked at a lot of exciting technology here, but the real focus for us today is how this applies to the IT industry. It's very easy to go out and get voice recognition technology, search and retrieval technology, as well as, the data mining technology.

But the question, as we brought forth earlier in this presentation is, it's not just using the technology, it's using the technology appropriately that separates the people who use customer service technology for the good as opposed to those who wind up misusing it and abusing their own customers.

When we take a look at the technology – we've already viewed a couple of people who are offering product suites in this area – we have to remember that they're relatively nascent newcomers to this type of technology area.

Only right now, in the year 2000, is Oracle 9i coming up with a new product suite called "Oracle Personalization," which offers a recommendation engine and a lot of the things we talked about in Web personalization.

The question for us in IT, however, is whether we want to buy a product suite to service our customers or whether we want to go with our own proprietary technology. It's the classic build versus buy issue.

Do we want the secret technology for competitive advantages, like we see in tools such as the Google search engine?

Or do we want an off the shelf, "it will do good enough," kind of product in order to get our customers to the customer support that they need.

We also have IT challenges regarding the integration of telephony and Web systems. Today we only see little less than 50% of the United States being Web-enabled, and a lot of people are still using telephones in order to query for customer service.

Given that the technologies of voice recognition are the same, it's entirely feasible that the backend systems can be entirely the same, whether the front end is a Web-based page or a telephony system accepting voice responses from an end user community.

We in IT have to be able to integrate these different diverse front ends into a common backend so that we can make all of these products look, feel, and act the same, regardless of where our customer comes to see us in the beginning.

After we deal with the integration of telephony and Web-based systems, we also have to remember that there are cases where the best placement of human intervention is appropriate in these kinds of technologies.

If we remember our example from Microsoft, having a human in there to tell individuals where they sit in a customer response queue gives our customers the kind of personalization that they feel that they deserve.

Remember, our overall goals are to make customers feel the same way they would if they were to come into a brick-and-mortar establishment, and have one-on-one contact with us.

This again ties into our review of personalization technology.

~ We have to have Web data tracking for customer warehousing, tracking both anonymous and

identifiable customers through the use of cookies.

- ~ We have to use warehouse classifications;
- ~ We have to come up with schemes for categorizing our products and categorizing our customers.
- ~ And then we have to create sophisticated statistics in order to come up with predispositions along those customer lines for what kinds of things they're willing to buy.

Again, there are off-the-shelf products in data mining to feed a recommendation engine, such as Oracle 9i's recommendation engine, that will do these kinds of product suites for you.

FOR ADDITIONAL INFORMATION

Once again, I'm Don Burleson. I thank you for watching the Core Technologies track, Personalization, Learning and Guidance Technologies. If you have a question or you'd like to make a comment, please e-mail me at don.burleson@watchit.com. I'll be glad to respond to your questions.