



From ReQueSt-DB to web-based search: An inventor's journey

Richard J. Cichelli
President, Software Consulting Services, LLC

When I was doing the research that led to my patent for distributing classified ads digitally, the issue of how to go from systems built for publishing in print to publishing in an electronic form was not far from my mind. As I saw the problem was how to solve the "ww" problem. What, you might ask, is the "ww" problem?

Patent number: 4429385 "Method and apparatus for digital serial scanning with hierarchical and relational access." Filed Dec. 31, 1981; issued Jan. 31, 1984 by inventors Richard J. Cichelli and Michael O. Thompson then with the American Newspaper Publishers Association.

Disambiguation

It goes like this - While reading a classified ad you find the abbreviation "ww" and want to know what it means. (No. It doesn't mean two-thirds of the start of a World Wide Web url.) There are "ww"s all over the place in classifieds. If the "ww" appears in a car ad, you assume it means "white wall tires." If it occurs in a residential dwelling ad, it probably means "wall to wall carpet." Likewise, if it appears in a personal ad, it means "white widow." Finally, if you are looking at seaside condominiums in Florida or California, it probably means "white water view."

Why would you care what it means? Imagine a search experience without such meanings. Your queries would yield few relevant ads and much unwanted data. Further, if you could really program a computer to recognize these abbreviations, then you could substitute "white walls" for "ww" in car ads before they are published. You would get better searches and more readable ad copy. (If the resulting copy is longer, you might even charge more to publish it. Wow!)

The computer science name for the "ww" problem is disambiguation. It means picking the correct meaning of ambiguous terms from context. Believe me when I say that this is a very big deal for the spooks down at the NSA.

Software Consulting Services, LLC

630 Selvaggio Drive, Suite 420

Nazareth, PA 18064

Sales: 1-800-568-8006

Fax: 610-746-7900

E-mail: sales@newspapersystems.com

www.newspapersystems.com

Better searching

In information retrieval (IR) science, providing better searches is measured in terms of recall and precision. Recall is a measure of how inclusive search results are. That is, did the search results include the information you are looking for. Precision is a measure of how well the search results exclude irrelevant results. (The folks who do bing.com claim to do this better than those at Google. You be the judge.)

Faceted search is the kind of search eBay provides. Searches of your online classified ads should be like eBay searches.

"Faceted search?" you say. Yes, "The overarching design goals of [faceted search] are to support flexible navigation, seamless integration of browsing with directed (keyword) search, fluid alternation between refining and expanding, avoidance of empty result sets, and at all times allowing the user to retain a feeling of control and understanding." - "Search User Interfaces" Marti A. Hearst.

So besides the ww problem, what's the issue of repurposing print classified ads for online search? When the purveyors of online classified systems show their wares, they assume a context that typically doesn't exist in your newspaper. To allow search for cars by mileage and/or price, they want you to fill in fields for these values when the ads are created. (It's why they say "Do online first.") This is fine for a self-service environment where this labor intensive process is done by motivated advertisers, but you surely don't want an employee ad taker doing this, right?

Patent Search Search

When I left the American Newspaper Publishers Association / Research Institute, I was presented with the rights to the patent. (Actually, the ANPA (now the NAA) left me and much of my staff when they moved to Reston,



VA in 1983.) We called the technology ReQueSt-DB for Relational Queries on Sequential DataBases. Bill Rinehart, then VP at the ANPA/RI, was thrilled with the technology, talked about it at ANPA/Tec trade shows and had played with a prototype for hours. (At first he proclaimed, as all who remember him knew he was wont to do, "No one will ever want to look at classifieds on a TV or VDT." (video display terminal) This was 1982.)

No one will ever want to look at classifieds on a TV or VDT.

Bill Rinehart, ANPA, 1982

Browse to

<http://www.aliciapatterson.org/APF001977/Fleischman/Fleischman04/Fleischman04.html> if

you want to get some insight into the joy of inventing the technology of the day. It was great to work at the ANPA/RI in the 70's.

The last NAA technologist

It is with considerable sadness that I note the recent passing of the last vestiges of the ANPA Research Institute. John lobst, PhD CS, my former student at Lehigh University, whom I hired in 1977 and who succeeded me when I departed in 1983, was let go by the NAA a few weeks ago. He was the last of the RI staff and the NAA's last technologist.

Classified search circa 1984

In 1984 as a new, struggling entrepreneur, I wanted to make a product out of ReQueSt-DB. I thought I needed three things.

- a way to deliver classified ads electronically as described in the patent,
- a way to turn print classifieds into ones suitable for electronic distribution (i.e., solving the disambiguation problem) and
- funding to develop the software systems needed.

For electronic distribution, the Lehigh Valley seemed ideal. Bark Lee Yee, a very early pioneer in the cable TV business, had a company that served the Valley. He had 43% penetration of households. (Believe it or not, his company, Twin County Cable, was number 2 here. TV reception in the Valley is terrible. Everyone had to have cable.) He offered to build microprocessor-based set top units that could support the ReQueSt-DB protocol at his Taiwan-based manufacturing plant. He would deploy these to Twin County subscribers. We would write the software following the algorithms in our prototype for the server and the set top units. I thought it was a good plan.

Delivery was simple - use a TV channel as a digital device and transmit character based ads at 6 megabaud. We built a 68000 based computer with 8 megabytes of RAM (huge then) and set it to transmit up to 8 megabytes of ads over and over every 10 seconds. It offered a 10,000 ad capacity with a typical retrieval time of less than a second. (Sub-second retrieval worked using intelligent frame prefetching.) In ReQueSt-DB the database rotated cyclically and every user's set top unit got to see it all once per rotation. It made cable into a general purpose, digital broadcast medium. Thousands of users could access the classifieds simultaneously with only a small constant load on a tiny (by today's standards) cable head-end server.

Named-entity recognition needed

Well, now that there is the World Wide Web, we aren't going to use that architecture. However, the parts of the problem having to do with categorization and disambiguation still remain. We called the tool for solving this part Convert-DB. Today you would think of this as a project to implement what's called the semantic web for classifieds. Believe it or not, computer scientists are working on this problem intensely right now. Most make something they think is really great, but don't like what their cohorts have produced so far. I join them and think they are all correct. The trick is they haven't seen what we have done.

Way back then I was looking for funding for this. A media mogul allowed me to visit him and some of his IT folks. I pitched ReQueSt-DB and Convert-DB. Before I tell you what he eventually said, let me assure you that you are going to learn how we solved this problem and made a technology that can take print classifieds from any system and turn them into a database which supports faceted search.

So here is basically what the mogul's technical guy said to me after he formulated his recommendations: "We have reviewed your technology and think there isn't much demand for electronic classifieds. Further, we think you are simply looking for funding." Well, duh! BTW - This same media mogul is still struggling with this issue. This was the last time we looked for anything like venture capital.

We have reviewed your technology and think there isn't much demand for electronic classifieds.

Media mogul, 1985

Fortunately, we participated in the Ben Franklin Technology Partnership, a state funded high-tec-company recruiting initiative. (SCS was the very first Ben Franklin company.)



They provided \$50,000 for each of two years for the development of Convert-DB.

A tools-based approach to software engineering

The way I design the software tools needed for such projects is to imagine the most expressive notation for programming their solutions. Then I write a formal language specification for the notation. SCS programmers then implement a compiler for this new programming language. You can't imagine how much better it is to program in a language custom designed for the specific task you face, rather than futzing with C, Pascal, C++, Java, etc.

Jonathan Gilman did the programming. He developed the compilers, the application, the prototypes. He was a truly gifted developer and a college dropout. I still have the program listings. The print-outs are six inches thick.

When Jonathan was done we could show how it worked and *still* I couldn't sell it.

At the same time our Layout-8000 (display ad dummyping) business was growing like gang busters. We created spreadsheets (VisiCalc) for conservative, mid and optimistic versions of our business plan, and we then proceeded to achieve the sum of all three! Sadly, we put ReQueSt-DB and Convert-DB on the back shelf.

Back to the future

So, last year we were building a web-based self-service classified advertising system to complement our interactive classified order entry system. We call this Community Advertising Services / Community Classified Services (CAS/CCS). Surely the next thing was to be able to search classifieds.

Who would have thought there was big money in search?

What a press is to print ads, search is to online ads. Web pages of search results can show paid classified ads that are the result of search queries as well as other ads, like banner ads, that are paid for and displayed too. (They are sort of an online version of classified display ads.) We call our solution for this Community Advertising Services / Community Classified Search Services (CAS/CCSS). (There's a theme here reflecting an engineer's approach to branding and marketing. Sorry.)

The patent teaches how to combine hierarchical searches (i.e., ones with categories, sub-categories and so on) with relational ones (i.e., those with ANDs, ORs and NOTs.) If you can recognize the attributes in the content of the ads while solving the disambiguation problem (called named-entity recognition) and map the results into searchable tags, you can set up a database for faceted search.

To make search easy to use, you need a user interface that supports the complex boolean expression queries of faceted search with point-and-click ease of use. Simultaneously, it would be nice if you could build the classification-specific attribute-oriented forms for web and interactive ad order entry with the IR and RDBMS updating software. A single context would support building software to enter, store and search.

What's the logic of search?

Boolean queries are tricky in and of themselves. Say you are looking for a house with enough bedrooms for your family of husband, wife and two kids. Would you say you want to look at ads for three and four bedroom houses? How many ads would you find with the boolean query "three bedrooms and four bedrooms?" The correct answer is none! No house is described as having a combination of three AND four bedrooms. What you want is a house with three OR four bedrooms. For someone who isn't a logician, common language does not a query make.

To solve this problem I created the notion of spanning sets. Bedroom attributes are a spanning set as are car makes, etc. Multiple attributes selected from a spanning set are formulated into queries with ORs. They are things of the same type. ANDs are used to combine attributes that are not part of spanning sets. ANDs combine things of different types. Air conditioning AND leather interior are attributes of different types.

So your search screens show categories and sub-categories etc. finally leading to a screen allowing the selection of multiple attributes within the hierarchy of classifications. Using the attributes selected and the declared spanning sets, a program can compose an IR query to feed to an IR database engine to search for the results. The result set returned is displayed as a faceted search. What you get might be a table listing showing a picture of the item, a brief description of it, a price etc. Just like eBay. Searches of your online classified ads should be like eBay searches.



ReQueSt-DB revisited

For a modern platform, we can use an information retrieval database and the Internet instead of a cable TV channel and a ReQueSt-DB set top unit. We chose the open source one from the Apache Foundation called Lucene. We built Lucene for IR indexing, storage and retrieval into our domain specific language (DSL), Spice. We thus combined both relational database management (RDBMS) and IR into a single integrated tool. (I believe such simple and complete integration is an active, open computer science research topic.)

Recognizer programs describe how ads are classified and tagged for hierarchical and keyword searches. They have sufficient context to provide the information needed to build the programs for data entry and searches.

(One question we often get is “What if a recognizer can’t figure out what the attributes of an ad are?” We call such ads “unclassified ads”.)

Rapid application development

SCS’s Bob Harwick implemented Convert-DB using our modern tools. Our tools for generative programming facilitated much of the work. Everything we implemented years ago was recreated even better and of deployment quality. Bob completed most of the work in just a few months.

The CAS team (Michael Grabowski, Dan Rinehimer and Joe Opsatnick) specializes in using our web development framework to build applications. They made a modern, web-based version of ReQueSt-DB in just a few months as well.

Even the recognizer programs themselves have significant parts composed by programs. One typical example is building the code to recognize car makes and models. Data entry programs written in Spice are used to enter make and model data into database tables. The Spice Macro Preprocessor uses macro templates and database access to emit the recognizer code based on the make-model table data. By such mechanisms sites can maintain their own make-model data and other similar lists of attributes.

Change the database, run the macro processor, get an updated more powerful recognizer. Compile the recognizer and feed it the ads. The output drives the database engine that supports search. It all works quite nicely.

With our tools, recognizers (that can process thousands of ads per minute) are easy to write and maintain. All the maintenance is in one context. From one terse specification, many software components are built.

The first recognizers we built for vehicles, real estate and employment were completed in less than one programmer month. This included the learning time and interaction with the tool developer.

Round two was in Spanish. In less than one and one-half weeks similar recognizers were built in Spanish by an individual who only had high school Spanish classes.

Conclusion

I hope you have found reading this light-hearted romp through newspaper technology as much fun as I had writing it. If you want to see what an automatically-built, web-based classified search engine looks like, follow <http://www.scssupport.com/demo> to one of our demos. From the menu bar across the top, select “Inquire” and then “Classified Ads”. Start by clicking on the “Search instructions” link on the right side of the screen.

Sample code from the recognizer for finding the number of bedrooms

```
.node realestate, attribute;

.sortkeys
! Beds is an integer of length 2
Beds: I: 2;

.spanningsets
! Define the attributes for bedrooms that will be matched
beds: "Bedrooms" {
  br1: "1",
  br2: "2",
  br3: "3",
  br4: "4",
  br5: "5+"
};

.recognize
! Store the number of bedrooms found in the sortkey "Beds" by
! looking for a number between 1 and 20 followed by "br"
Beds: 1, =1::20 "br";

! Now use the number stored in the sortkey "Beds" to determine the
! correct attribute in the "beds" spanning set
br1: .accept { `(Beds = 1) };
br2: .accept { `(Beds = 2) };
br3: .accept { `(Beds = 3) };
br4: .accept { `(Beds = 4) };
br5: .accept { `(Beds > 4) };
end
```

