

Preface

Adding CBR Back into the Lexicon: Today's CBR Technology Not Only Works, It Makes Strategic Sense

John Ragsdale, Giga Research Director, Forrester Research

Let me start by stating: case-based reasoning is not evil.

This may come as a surprise to those of us who have been in and around the world of CRM (customer relationship management) and technical-support knowledge-base products for a number of years. I personally have been guilty of associating “cased-based reasoning” (CBR) with systems that are hard coded, inflexible, difficult and expensive to maintain, and hard to use effectively.

But it is important to separate the concept of CBR from its (supposed) instantiations. Early knowledge-base products created for technical-support agents in the late 1980s and early 1990s were sometimes billed as CBR. The first such systems I encountered took a year to implement in my support center (all custom work by a systems integrator [SI]), and a week after go-live my agents staged a walkout. Either the system went, or they did. Talk about low-adoption rates.

Many support managers were burned by systems sold as “CBR,” and I have encountered quite a few other ex-technical support or helpdesk managers, some of them now VPs of support or even at the CIO level, who have a visceral reaction to the term “CBR.” Their reaction is often a surprise to software developers and those from the more academic side of knowledge management.

So before you begin this truly insightful and promising book about case-based reasoning and learn how to more effectively access and leverage corporate knowledge, let's clear up what CBR *is not*:

- CBR is not a set of solution documents organized in a hardcoded index “tree.”
- CBR does not force agents or customers to go step by step down a predefined path through the minutia of choices that never exactly fit the situation or problem.
- CBR does not mean a single static solution to a single static problem that requires a programmer to add, change or delete.

Having worked with and for software vendors selling intelligent knowledge-management products geared toward technical support agents and/or customers performing Web self-service, I've often advised against using the term “CBR” for

fear of backlash from prospects, who like me, may have been burned by a really bad implementation sold as CBR.

But after reading the first edition of this book, and talking with Michel Manago, one of the co-authors, about what CBR actually means, I think it is time to revive the CBR term. It's also time to stop punishing the entire CBR concept for the sins of a few early software vendors and greedy SIs. And it's time to move forward with learning how to use CBR technology to streamline customer service operations, cut the operating costs of technical support centers, and offer stellar web self-service options to customers who have the need, the desire, or just the plain old *moxie* to want to figure things out on their own.

The Business Case for Easier Access to Knowledge

In most customer service centers, customer interactions are increasing as much as 30 percent year after year, counting all of the new “e” interaction channels, such as e-mail and chat, in addition to phone calls. The percentage of interactions by channel is changing as well, as more customers adopt electronic channels. While the volume of phone calls received by the average support operation may remain the same, or increase slightly as other interaction channels grow in popularity, phone calls constitute a smaller percentage of overall interactions. E-mails are expected to bypass phone calls in most industries by 2010.

All of this is forcing companies to fund initiatives for customer self-service, hoping to entice more and more customers to attempt self-service before calling for assistance or at least to streamline agent processing of customer questions with robust knowledge-base options. Ideally, these scalable solutions allow companies to compensate for increases in interaction volume year after year by minimizing both the number of interactions involving agents and the time agents spend processing each interaction.

Expecting even a trained agent (let alone a novice customer) to find useful information in a static knowledge base by using only a full text search is unrealistic. Customers cannot possibly know all of the key “symptom” words to include, and the more textual clues the customer provides in the search string, the larger the number of possible matches returned. Most customers attempting self-service will abandon all hopes of resolving their own problems when the text search returns over 1,000 possible matches. And such results train customers to never attempt self-service again, forcing them to migrate to the most expensive support channel: a live phone call with an agent.

Case-based reasoning makes sense in those industries where internal or external customer support involves disparate or complex hardware or software configurations, multiple technical environments or platforms, and varying levels of service or warranty options, as well as vastly different experience levels across the enterprise or customer base.

Why CBR Makes Sense Now — Technically

CBR solutions have credibility today because three main technological areas have come a long way since the 1980s: linguistic analysis, consolidated customer databases, and intelligent analytics.

Natural Language Searching (NLS) comes in many flavors, far too many to describe and differentiate here. But one area of NLS, linguistic analysis, has moved out of the laboratory and into packaged software products, allowing a system to intelligently understand what a customer (or agent) is asking, not just to perform a keyword search. It's also about attribution, i.e., the amount of ancillary data that can be factored into a problem diagnostic session to more succinctly identify the correct answer or knowledge "chunk." Before the days of XML and SOAP interfaces, only minimal data was passed between systems, and few environmental variables were stored in a single database.

CRM has changed all that. A single CRM customer-information model can create a "360-degree view" of the customer, listing every product the customer has purchased, the technical environment in place (Unix vs. Linux vs. NT, or Oracle vs. Sybase vs. SQL), every problem the customer has experienced in the past, and any other information a company may choose to capture, such as the user's skill or training level, preferred language, etc.

With this amount of attributable information at the disposal of a CBR application, customers (or agents) need not manually supply background information (or follow an index tree answering basic questions). The *context* of each question is automatically established, allowing a CBR system to more granularly identify a case that meets the exact question in relation to all environmental variables. The more the system is used, and the more users indicate which solutions solved which problems, the more environmental data from all of those users can be captured within the case framework, thus allowing the system to continually "learn." Add to this the advent of analytic engines that can make intelligent, real-time and even personalized decisions and recommendations based on any amount of available data, past history, and any rules in place. Though both software vendors and IT developers are wary of the term "artificial intelligence" (AI) — often because they were burned by early CBR systems that were said to use AI but failed to deliver — real-time analytics are here, and vendors have customer references to prove it.

A lot of things have changed since the mid – 1980s. Giving agents tools to increase productivity and offering robust customer self-service systems is no longer an academic exercise; it is the only way large support operations can even come close to meeting customer expectations in this age of budget cuts and staff downsizing. And today's CBR-based products don't take a year to implement; they can guarantee a positive return on investment in as little as three months. For many support environments, CBR is a logical choice.