# Intelligent Electronic Catalogs for Sales Support

## Introducing Case-Based Reasoning Techniques to On-Line Product Selection Applications

**Ivo Vollrath, Wolfgang Wilke, and Ralph Bergmann**

*Department of Computer Science*
*University of Kaiserslautern, PO-Box 3049*
*67653 Kaiserslautern, Germany*
*http://wwwagr.informatik.uni-kl.de/*

**Keywords:** Electronic Catalogs, Sales Support, Case-Based Reasoning, Electronic Engineering, Databases

## Abstract

*The number of electronic catalogs has grown rapidly during the past few years. Most of these catalogs use standard databases for storing and retrieving product information. Using ordinary databases for product catalogs, however, has the major drawback that it is often very difficult to find the products desired: very often, the database does not return a matching product at all or it returns many products that have to be examined manually. To overcome this problem, we propose the use of Case-Based Reasoning (CBR) techniques as an approach to requirement-oriented retrieval of products. CBR incorporates product knowledge into the database by means of a* similarity measure. *Recently, a number of commercial electronic catalogs based on CBR have been realized. We take a closer look at Analog Devices' on-line catalog of operational amplifiers, which helps an engineer to find a suitable amplifier for her specific requirements.*

## 1 Introduction

The number of electronic catalogs – especially on CD-ROMs and on the Internet – has grown rapidly during the past few years. In certain areas they even have become as important as their printed counterparts. This is especially true for electrical engineering applications where many vendors offer product information over the Internet and most of their customers use this service frequently. Most of these catalogs employ standard database approaches for storage and retrieval of product information. While today, databases are well understood and many off-the-shelf solutions exist, we claim that they can only be used efficiently by advanced users – users that are already very familiar with the contents of the database. This is especially a problem for electronic catalogs meant to be used directly by end consumers or engineers who, in many cases, are not experts in the family of products they are looking for, or at least do not have a good overview over all the products and their specialties inside the catalog. Imagine an electronics engineer looking for a device to integrate into her new circuit that pushes the limits of the most advanced devices from her catalog. How can this person select the device *best fitting* her needs if she is no expert in the area of devices she is looking for? A standard database offers no support for her.

In this paper we give examples of how difficult it is often to find desired product information in electronic catalogs. We introduce the basic idea of Case-Based Reasoning (CBR) and discuss the possibilities that this technique offers to overcome many of these problems. A number of commercial applications using part of the proposed techniques recently have been realized[1]. Information brokerage in high precision manufacturing and electronic parts catalogs are covered by these applications. We provide a case study of a commercial application. Analog Devices' on-line catalog of operational amplifiers is presented in detail and we close with a short discussion.

## 2 The Problem with Ordinary Databases

So why should a customer searching for product information on the web not be satisfied with a huge database of products and a database search engine on top of it? The most important answer to this question is: "Because this system entirely fails to provide *sales support.*" We illustrate this problem with the following scenario: A customer enters a shop that sells electronic parts. Imagine the shop assistant acts like a database search engine:

**Customer:** Hello, I need this transistor 'POC-4711'.
**Assistant:** Sorry, we currently do not have this transistor in stock. *(No further explanation, no alternatives.)*
**C:** Er, well, do you have the military version 'POC-4711-m', then?
**A:** Sorry, we do not have this one in stock either. *(The Customer probably leaves the shop now, but let's see what happens if she doesn't.)*
**C:** Well, do you have any transistor coming close to the POC-4711's amplification?
**A:** Can you please exactly state what you mean by saying 'close'? Can you specify a range?
**C:** (Sigh!), let's say 30 to 35.
**A:** Sorry, we do not have any . . .
**C:** *(Leaving frustrated, looking for another shop.)*

Now let's see how a real shop assistant would have managed the situation:

**C:** Hello, I need this transistor 'POC-4711'.
**A:** Sorry, the 'POC-4711' is out. But I can offer you the 'BAC-42'. It behaves quite similar.
**C:** What is it's amplification?

[1] At http://wwwagr.informatik.uni-kl.de/˜lsa/CBR/wwwcbrindex.html you find a collection of links to CBR based online product catalogs.

**A:** 37.
**C:** *(Will probably buy the 'BAC-42', but let's see what happens if she doesn't:)* Hm, I don't know… I might get into trouble with that.
**A:** Okay, why don't you connect two 'YH-12'? They will do the job.
**C:** *(Satisfied, buys two YH's.)*

Finding the desired information or product in an online catalog can be rather frustrating given the standard database implementations that are commonly used today. With standard databases it is very difficult for the occasional user to "balance" a database query in order to not produce too many results – or just no answer at all. A customer using some kind of search facility to find a product in an online catalog must fulfill certain requirements in order to have a chance to get usable results: If she knows the product's part number in advance she will certainly get the desired product information. If the customer is not looking for a specific part number but for some product that *has certain qualities*, she is in a much more difficult situation. In that case she must be very familiar with the contents of the database or otherwise she will get either no answer to her query or a whole bunch of answers that might not even be sorted in any usable way and therefore do not help at all.

By incorporating product knowledge into the database CBR is much more flexible and powerful than by just offering menu driven product selection or a searchable database. In this paper we present a scenario of how *similarity based retrieval* can partly emulate a shop assistant and offer real *sales support* to the end consumer. A CBR system's product knowledge enables it to judge which products optimally fit the customer's needs. CBR systems with configuration capabilities can even *adapt* a product to the user's demands which otherwise would not satisfy her requirements.

The fact that today's standard online catalogs cannot offer intelligent sales support requires of the customer that she is a semi-expert in the field of products she is interested in, in order to solve this problem. It is very important to put the knowledge of the shop assistant into the retrieval component of the online catalog. The system must have enough *domain knowledge* about the products to be able to aid the customer in her search. This would not only satisfy the customer, but it would also help the manufacturer or broker in selling her products. In Sect. 4 we will see how Case-Based Reasoning can help to overcome these deficiencies.

## 3 Extended Search Approaches

Currently there exist a number of systems that try to help the user searching for certain information or a certain product on the web in an intelligent way. Well known types of systems are meta search engines as well as search agents.

But these systems do not offer the kind of sales support this article is focused on. This is because they have no knowledge to differentiate between *similar* products of the same type and what their difference of value is to the user. Especially, query routing systems mainly rely on the intelligence of the engines the query is routed to and have only little additional knowledge of their own.

Often a customer needs support in selecting a product that best meets her requirements from a catalog of many similar products. This kind of sales support (or *decision* *support*) requires knowledge on a much lower level of abstraction than search agents can provide. A conventional printed product catalog offers only little sales support to the customer but sometimes there is extensive information about how one can find the best product for her needs. If you ever tried to select a mountain bike from a catalog you know what we are talking about: *How do I determine the right frame size for me? What kind of wheels do I need? What are break boosters good for? Do I need them?* etc. It usually takes a lot of time to read all the additional information and to decide for a certain product. Interactive technologies offer much more possibilities to overcome these problems and prepare for a much better substitute for a shop assistant than printed media do. With a good interactive catalog with sales support capabilities the customer should not need to become an expert in the type of product she is looking for.

In general, there are two opposing ways to design interfaces for such catalogs. The *product-based* approach offers support for guided navigation through the product space. This can be compared to finding a certain product by searching through a highly structured table of contents of a printed catalog. With this approach the customer step-by-step is lead through the decision process. To stay with our mountain bike example, she might first be asked for her size and weight, then for the intended use of the bicycle (off-road, city, etc.) and so on, until the system finally suggests a small number of bikes suitable for her. In more complex applications the fixed order of decisions can become a major problem. Questions the user cannot answer at a given time are also difficult to handle. After all, it is a common experience of people doing after sales support using such a system that they find it very tiring having to go through all the presented questions and selections.

The *requirements-based* approach allows the user to specify all or part of her requirements by using some (more or less pre-structured) query form. The simplest facet of this kind of systems are searchable databases where the user specifies the demanded attributes of a product. This allows for very flexible queries but as we have seen before it is only useful for expert users who are familiar with the contents of the database.

Case-Based Reasoning offers solutions for requirement-based query interfaces that preserve the flexibility of searchable databases but integrate the advantages of product-based approaches and make them useful for both first-time users and experts. By incorporating product knowledge into the database it is much more flexible and powerful than just offering *both* menu driven product selection *and* a searchable database.

## 4 The Concept of Case-Based Reasoning

The basic idea of Case-Based Reasoning (a good overview is given in [2]) is to solve new problems by comparing them to old problems that already have been solved in the past (Fig. 1). The key assumption is that if two problems look *similar*, then the solutions to these problems are often similar as well. However, CBR systems must know how to decide whether two cases are similar or not.

Old problems and their solutions are stored in a database of *cases*, called the *case base*. Often the cases are stored as collections of *attribute-value pairs* but sometimes it is useful to explicitly represent the hierarchical structure of the cases by describing them as structured objects, using inheritance, object decomposition and possibly other relations between the object's parts.

When a new problem has to be solved, the CBR system searches for the most similar old problem. The solution of this old problem may optionally be *adapted* to better meet the requirements of the new problem. CBR systems have been successfully used for various applications such as fault diagnosis of complex machines, classification, decision support, support for planing, configuration and design tasks, legal reasoning and various other application areas [3]. Another commercially interesting field where CBR is widely used and has proved its qualities are help desk applications. For further links to information on CBR applications and its theory see the CBR homepage at http://wwwagr.informatik.uni-kl.de/~lsa/CBR/.

In our application scenario the cases are descriptions of the products to be sold. The problem description is a specification of a single product and possibly demands that are satisfied by it. The solution to the problem is an unambiguous reference to the product. For configurable products such as workstations, automobiles, complex machines, etc. the solution is not only the part number, but possibly the entire configuration. When a customer enters a query (perhaps into a query form), this query is regarded as a new problem and the CBR system tries to solve this problem by comparing it to the cases in the case base.

Imagine an online catalog of holiday accommodations: When someone wants to spend her holidays on the beautiful island Pariku in a 3-star hotel, her preferences are interpreted as a problem description. The CBR system will then find one or more similar cases which best match the query. There might be no 3-star hotel on Pariku, so the system might suggest a 2-star hotel on Pariku and alternatively a 3-star hotel on the neighbor island Pung-Chitu.

We described the important roles of the similarity measure and the retrieval and adaptation components of a CBR system. We will now take a closer look at these aspects.

### 4.1 Similarity Measures

A frequently asked question is "What makes Case-Based Reasoning different from ordinary database systems?". Perhaps the most important answer to this question is: "CBR

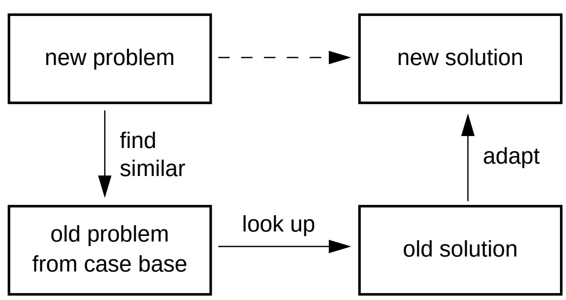


**Fig. 1.** The idea of CBR is to solve new problems by reusing the solutions of old ones.

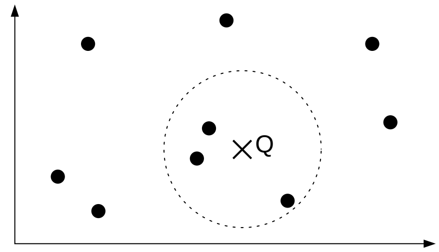


**Fig. 2.** Usage of similarity in CBR: The cases are distributed over an $n$-dimensional problem space. The circle around the query Q indicates a similarity threshold for the cases to be retrieved.

systems have additional domain knowledge built into them." The main part of this additional knowledge is implemented in a *similarity measure* which is a function that assesses the similarity of a given query to the cases in the case-base. The similarity values are ordinal values that are often normalized to the interval $[0, 1]$. A value of 0 means "does not satisfy the query at all" and a value of 1 says "that's exactly what you asked for".

To understand how such a similarity measure is used to find the best solutions for a given problem consider the cases being represented as a fixed length vector of $n$ attributes. These attributes may have numerical values or their values can be arranged to reflect some kind of order. Part of the description of an electronic device, for example, might be:

| | | |
|---|---|---|
| case1 = ( | 5, | *supply voltage [Volt]* |
| | 0.25, | *max. power consumption [Watt]* |
| | 32, | *number of pins* |
| | . . . ) | |

The problem space can now be seen as an $n$-dimensional space where similar problems are placed 'close' together (Fig. 2). The term 'close' is defined by the similarity measure. When a new problem is presented to the system, the similarities to all the cases in the case base are calculated (the number of visited cases may possibly be optimized by suitable indexing techniques). The cases within a similarity range of a certain threshold (or sometimes a fixed number of similar cases) are then presented to the user.

But how can such a similarity measure be defined? The usual approach is to start with individual (local) similarity measures for all the attributes. For numerical attributes a natural approach is to calculate the difference between the query value and the case's value and normalize the result to the interval $[0, 1]$, for example by applying the formula $1 - \frac{d}{d+1}$ to the difference $d$. But certain attributes require specialized functions. Imagine, for example, our description of electronic devices. It is often (but not always) the case that the tolerances for the supply voltage of a device are much tighter than the user's tolerances for the device's max. power consumption. The similarity function must reflect this domain dependent knowledge.

In real-world applications not all attribute values are numerical values. A case description very often contains boolean or symbolic attributes. In such cases, the local similarity measure can be described by a table which defines the similarities for all possible pairs of attribute values (Fig. 3). Even more complex attribute types such as taxonomy types or complex objects sometimes are required.

After the local similarity functions have been defined, the global similarity of two cases must be derived from the local similarities (and possibly from additional information like constellations of certain attributes). The usual way to do this is to apply a weighted sum to all the local similarities. Consider a query $q$ is described by the attributes $q_1, \ldots, q_n$ and a case $c$ is described by $c_1, \ldots, c_n$ where attribute values with corresponding indices belong to the same attribute. The similarity $\sigma$ between $q$ and $c$ can be calculated from the local similarities $\sigma_i$ as follows:

$$\sigma(q,c) = \sum_{i=1}^{n} w_i \sigma_i(q_i, c_i) \qquad \text{where} \quad \sum_{i=1}^{n} w_i = 1 \quad \text{and} \quad w_i \geq 0 \text{ for all } i$$

The weights $w_i$ have to be assessed by an expert but sometimes it is useful to give the user the possibility to manipulate them in order to express individual preferences. This general method of defining a similarity measure as a weighted sum can be modified and refined in many ways. In general the computation of the similarity measure depends very much on the data model and on the type of application.

Finding a suitable similarity measure often is the most critical part during the design and implementation of a CBR system. Once a good similarity measure has been found and implemented, maintaining a CBR system is rather easy. The similarity measure typically doesn't need to be changed for a long time compared to the knowledge bases of other types of expert systems. Often the only part that undergoes significant changes over time is the case base, whereas the domain-model and similarity measure rarely need to be changed.

### 4.2 Retrieval Techniques

We said before that, in order to find the cases that best satisfy the query, the similarities of the query to all cases from the case base have to be calculated. But it is not always necessary to do a linear search which, of course, would be a time-consuming process. To significantly reduce the retrieval times of CBR-Systems a number of indexing techniques have been developed. Generally it is difficult to adapt standard database indexing algorithms to similarity based systems. This is because these database approaches do not take into account soft similarity values, but only support hard selection criteria.

One example of efficient indexing structures are *k-d trees* [4] which are modified versions of decision trees that allow to significantly reduce the number of visited cases while soft similarity values are still taken into account. Another approach to fast case retrieval are *case retrieval nets* [5]. The idea is to represent the cases and their attributes as a network of interconnected *information entities.* Starting with the query's information entities activated, a spreading activation algorithm is used to retrieve the best matching cases. Case retrieval nets are successfully used in some existing on-line applications for travel agencies.

Both of these indexing approaches significantly speed up the retrieval times by *compiling* the similarity measures into indexing structures. They become inefficient if the case-base needs to be updated very frequently, i. e., if the number of updates is in the order of the number of read accesses.

### 4.3 Adaptation

Another point that makes CBR different from standard database approaches is its ability to not only retrieve existing cases, but also *adapt* their (old) solution (here: the product that is to be sold) to the newly presented problem and thus create new solutions that were not represented in the case base [6]. This requires, of course, that the solution is configurable. Let's look at a simple example: Imagine a car dealer who runs a CBR-based catalog for her stock of used cars. A customer who is looking for a used car enters her preferences into the retrieval system. One feature that the car should have is a sliding roof but the dealer's stock does not contain any car with a sliding roof. A CBR system with adaptation capabilities will know that a sliding roof can be built into a car later. So the system will automatically adapt the descriptions of the best matching cars to contain an entry saying that there *is* a sliding roof. The price for the sliding roof and the wage for its installation is automatically added to the 'price'-attribute. This happens totally transparent to the user.

The knowledge needed to perform such kinds of adaptation must be represented in some suitable form. One possible representation is a set of *rules* that perform certain actions given that the required preconditions for these actions are valid. A rule for our sliding roof example could be written as follows:

```
if   query.slidingroof == true and
     case.slidingroof == false and
     case.cabriolet == false
do:  case.slidingroof := true;
     case.price := case.price + $200
```

Depending on the application domain, the adaptation process can be more or less complicated. To our knowledge there is currently no commercial CBR-based product catalog that supports adaptation. The reason for this probably is that new technologies are always first applied to simple problems and only later to more difficult tasks. Also, there are many products that cannot be adapted in any way because they have no modifiable structure. On the other hand, adaptation is indispensable if the system should be able to perform some kind of configuration task, like interactive configuration of personal computers. The WEBSELL project described in Sect. 6 works on adaptation techniques for CBR-based on-line applications. [7] gives an overview of configuration technologies based on CBR.

### 4.4 Advantages

Let us summarize some of the advantages of CBR-based product databases over standard database techniques:

– Near misses are avoided: If the customer asks for features that cannot be fully satisfied by products from the database, she is still offered something *close* to her request.

- The fact that there is an answer to almost any query greatly reduces frustration on the customer's side and helps the vendor in selling *nearly perfect* solutions.
- The solutions are ranked by their similarity to the query. This is essential when the system finds dozens or hundreds of solutions. In opposition to other retrieval systems (such as some Internet search engines) this ranking is not only based on some statistical frequencies but it is based on *knowledge* that has been acquired from real experts.
- The adaptation process can even *create* solutions not contained in the original database. Thus CBR offers support for interactive or automatic product configuration.

## 5 Existing CBR Applications

The number of online catalogs with built-in CBR technology is yet rather small. Nevertheless, the quality of their search results are very promising. We encourage the reader to check out the online versions of the products described at:
http://wwwagr.informatik.uni-kl.de/~lsa/CBR/wwwcbrindex.html

### 5.1 A Case Study: Analog Devices

We will now have a close look at Analog Devices' online catalog of operational amplifiers. Some background information: Analog Devices is one of the major manufacturers and sellers of electronic devices in the U.S. According to the company's annual report [8], the company has direct sales offices in 17 countries. Many of Analog Devices' customers are small electronics firms or electronics departments of larger firms. Analog Devices currently employs a number of engineers to provide central applications support for their customers. There is also an online catalog with certain standard search capabilities and selection trees. Many people calling the hot-line do so because they could not find a suitable match for their problem in the printed or online product catalog. Roughly half of the calls to the hot-line support relate to product selection. It is obvious that the number of calls to the selection support service could possibly reduced significantly, if only the online catalog was able to provide intelligent sales support. So it has been decided to offer CBR-based search facilities in addition to the standard database interface. In a first stage the application domain for this system was chosen to be Analog Devices' product family of operational amplifiers, but the system is currently being extended to cover at least three more product families. Their product catalog contains about 130 operational amplifiers suitable for a wide range of applications. In many cases the new CBR system is able to support the customers in selecting a device that satisfies their requirements. This reduces the number of calls to Analog Devices' hot-line and enables some engineers to care about more advanced support problems that cannot be solved automatically.

#### Case Representation

The data sheet for an operational amplifier consists of about 40 parameters which can be logically structured into categories of parameters, such as electrical input and output specifications, functionality, dimensions, etc. Most of the parameter values are high exponent real numbers, but there are also symbolic and string parameters. No object decomposition or hierarchical representation has been used for this application. The retrieval interface provides hyper-links to explanations of each of the 40 attributes.

### The Similarity Measure

To assess the similarity of two operational amplifiers, the similarities for all corresponding parameter values are calculated by applying local similarity functions to each pair of corresponding parameters. These local parameter similarities are then used to calculate an overall similarity value for the two devices. The overall similarity is computed as the weighted average of the individual similarities. The original weight factors have been suggested by experts in operational amplifiers but they can be influenced by the customer according to her priorities. The local similarities for discrete and continuous values are calculated in different ways.

*Discrete* similarity measures are defined by a table which explicitly lists the similarity values for all possible attribute combinations (Fig. 3). Note that the table is asymmetric: it makes a difference whether the query's attribute value is $x$ and the case's value is $y$ or vice versa. If, for example, the user asks for a device with industrial standard temperature specifications, she will be satisfied by a part that fulfills military requirements (provided that no other attributes, such as the price, stand against it). On the other hand, the request for military standards can not fully be satisfied by a part that only has industrial specifications. This situation is reflected by the asymmetry of the similarity table.

This principle also holds for most of the *continuous* attributes of an operational amplifier. The only difference is that their similarity measures cannot be as easily represented in a table, but must be formulated as a function. An example of one such similarity function is shown in Fig. 4. This function assumes that there is a minimum and a maximum attribute value. If the case's attribute value is less than or equal to the query attribute's value their similarity is 1. Otherwise it is a value somewhere between 0

| query \ case | com. | industr. | military | space |
|---|---|---|---|---|
| com. | 1 | 1 | 1 | 1 |
| industr. | 0.6 | 1 | 1 | 1 |
| military | 0.4 | 0.6 | 1 | 1 |
| space | 0.1 | 0.4 | 0.6 | 1 |

**Fig. 3.** An example similarity function for discrete attribute values taken from Analog Devices' operational amplifier application. This is the (simplified) similarity function for the parameter "maximum temperature range" of an operational amplifier. The symbols 'commercial', 'industry', 'military' and 'space' describe temperature range standards of electronic devices. If, for example, the query specifies 'industrial' standards, the CBR system regards 'military' and 'space' to fulfill the requirements, but 'commercial' has a similarity of only 0.6.

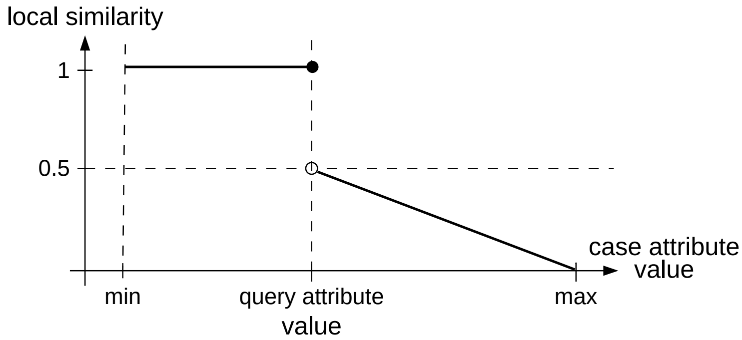


**Fig. 4.** Another example similarity function but for continuous attribute values. The similarity function for the parameter "current noise density" of an operational amplifier is shown. The corresponding attribute of an old case has similarity 1 if its value is less than or equal to the query's value. Otherwise the similarity is significantly smaller.

and 0.5. For certain other attributes the function must be reversed to return 1 for values greater than or equal to the query's value and 0 to 0.5 for smaller values.

This application does not use adaptation. This is because operational amplifiers are unchangeable parts that cannot be reconfigured to the customer's individual demands.

### The Benefits: Sales Support

A typical sales support session goes as follows: The customer enters the parameter values she needs into the query form. The system will then retrieve the ten best matches to the request. If the results do not exactly fit the customer's needs, she will usually increase the priorities of the parameters that are most important for her. Again, the system displays the ten best matches to the refined query. If the results still do not satisfy the customer, she might fill more parameter slots that she left empty so far, thus further improving the quality of the returned results. When finally a suitable device has been found, the customer can link directly to its detailed data sheet.

This is very similar to a real shop situation when a customer consults the shop assistant: The assistant step by step learns about the customer's exact demands until finally she finds the product that best fits the customer's needs. A CBR system has part of the knowledge of a real shop assistant built into it. This knowledge is used to interpret the user's query and greatly enhances the quality of the retrieved data. Even the first request immediately produces usable results. Near misses are avoided. If the system returns more than one answer, the results are ranked by their similarity to the query. This makes it easy for the user to refine her request until she is satisfied with the results. This is why we claim such a system being capable of intelligent sales support.

But not only the customer gains profit from this intelligent sales supporting catalog. As this approach will be pursued further, Analog Devices in the future will be able to allow some of their engineers to spend more time addressing more complex customer support requirements. Encouraged by the first success of the new CBR catalog, Analog Devices have decided to further extend the application of CBR technology to other product families. Work on these follow-up projects has already started by time of writing of this article.

## 6 Current and Future Work

The research project WEBSELL which has been started recently has the primary goal to explore techniques to support this sales process especially by employing Case-Based Reasoning and related methods. WEBSELL is a research and development project funded by the Commission of the European Communities (ESPRIT contract P27068, the WEBSELL project: Intelligent Sales Assistants for the World Wide Web) with the partners: TecInno (Germany, prime contractor), Adwired (Switzerland), EuroWEB (Germany), Irish Medical Systems (Ireland), Trinity College Dublin (Ireland) and the University of Kaiserslautern (Germany).

Another new research project recently has started with the aim to further explore and integrate the possibilities of CBR-related techniques into *reuse* applications and product catalogs. The READEE project[2] (*R*euse *A*ssistant for *D*esigns in *E*lectrical *E*ngineering) employs CBR-techniques to re-use designs of electronic circuits. Reusing such designs in general can require solving very complex configuration tasks. The reuse assistant is meant to help electrical engineers in selecting and configuring third party intellectual property to fit their personal requirements. Making such a reuse tool available over the Internet or on intra-nets of larger electronics companies can speed up the time-to-market of new electronic circuits significantly. Both the vendor and the consumer of the intellectual property benefit from this tool. READEE is funded by the Foundation for Innovation of Rhineland Palatinate and will last for a period of three years.

When interactive electronic sales support will become more common in future, interfaces for interactive adaptation and configuration will come into use. CBR-related techniques for indexing and clustering the case base can be used to help the customers refine their queries by a step-by-step analysis of their needs. By analyzing the structure of the case base, a CBR system can suggest which undefined parameters the user should define next in order to find a good solution as quickly as possible. It is also possible to *explain* to the customer the reasons why the retrieved results are suitable for her query. For instance, the previously mentioned example online catalog of holiday accommodations might tell the user who requested a 3-star hotel on Pariku: “I suggest this 3-star hotel on Pung-Chitu because Pung-Chitu is close to Pariku and has a similar landscape and infra structure.”

## 7 Conclusion

CBR technology introduces intelligent sales support to electronic commerce applications. The deficits of standard database techniques are overcome by adding expert knowledge to the retrieval system. CBR gets rid of the problem of near misses, the need of expert knowledge on the customer’s side and resulting frustration. New solutions can be created by adapting old solutions, as we saw in our sliding roof example

[2] http://wwwagr.informatik.uni-kl.de/˜readee/

for the catalog of used cars – and even complex configuration tasks can be simplified by Case-Based Reasoning. Finally the vendor benefits from the ability to offer nearly perfect solutions and from satisfied customers that do not need to look for other vendors any more just because the database search engine could not find a product matching the customer's requirements. There are already many positive experiences with currently available applications. New technologies can be expected to offer sophisticated adaptation and interactive configuration abilities for complex products. This leads to negotiation between the virtual sales agent and the customer on the Internet [9]. Interactive refinement of the customer's requirements and explanation of results are also possible. Some future improvements to intelligent sales support systems are current research topics.

## References


1. Doorenbos R B, Etzioni O, Weld D S 1996 A Scalable Comparison-Shopping Agent for the World-Wide-Web. *Technical Report UW-CSE-96-01-03*. Department of Computer Science and Engineering. University of Washington
2. Leake D 1996 CBR in Context: The Present and Future. In: Leake D (ed) 1996 *Case-Based Reasoning, Experiences, Lessons, & Future Directions.* MIT Press, Menlo Park, Calif, pp 3-30
3. Althoff K-D, Auriol E, Barletta R, Manago M 1995 A Review of Industrial Case-Based Reasoning Tools. *AI Intelligence*
4. Wess S, Althoff K-D, Derwand G 1994 Using K-D Trees to Improve the Retrieval Step in Case-Based Reasoning. In: Wess S, Althoff K-D, Richter M M (eds) 1994 *Topics in Case-Based Reasoning.* Springer Press
5. Lenz M, Burkhard H-D 1996 Case Retrieval Nets: Basic Ideas and Extensions. In: Görz G, Hölldobler S (eds) 1996 *KI-96: Advances in Artificial Intelligence.* Springer Press
6. Wilke W, Bergmann R 1998 Techniques and Knowledge used for Adaptation during Cased-Based Problem Solving. On the *11th International Conference On Industrial and Engineering Applications of Artificial Intelligence and Expertsystems (IEA-1998).* Springer-Press
7. Wilke W, Cunningham P, Smith B 1998 Using Configuration Techniques for Adaptation. In: Burkhard H-D, Bartsch-Spörl B, Lenz M, Wess S (eds). *From Foundations to Applications in Case-Based Reasoning.* Springer Press
8. The Board of Analog Devices 1995 *Analog Devices Annual Report 1995.* Analog Devices
9. Wilke W, Bergmann R, Wess S 1998 Negotiation During Sales Support with Case-Based Reasoning. In: *Procceedings of the German Workshop on Case-Based Reasoning (GWCBR)*