8.1 USER-CENTERED DESIGN IN CHEMICAL EMERGENCY SOFTWARE

B. Connolly*
I.M. Systems Group, Inc., Rockville, Maryland

D. Wesley and M. Miller
National Oceanic and Atmospheric Administration, Office of Response and Restoration,
Emergency Response Division, Seattle, Washington

1. INTRODUCTION

When a hazardous chemical is released—either deliberately or accidentally—decision makers need to determine quickly how best to respond to the situation in a manner that minimizes the impacts to human health and safety. To make these operational decisions, emergency responders need information that is timely, accurate, and easily understood. CAMEO (Computer-Aided Management of Emergency Operations) is a suite of software programs designed to assist decision makers quickly get the information they need for a safe, effective response. (The CAMEO software suite is developed jointly by NOAA's Office of Response and Restoration and EPA's Office of Emergency Management.)

In order to create software that works well for users in high-pressure, emergency response situations, the CAMEO development team has gravitated to a technique called User-Centered Design (UCD). For over 20 years, the development team has been interacting directly with emergency responders, planners, and forecasters to ensure that a simplified, unambiguous user interface is coupled to the suite's chemical database and complex air dispersion model to allow a widely disparate user community access and understanding to the critical emergency response information.

2. USER-CENTERED DESIGN (UCD)

UCD is a development technique that designers can use to create products that people describe as user-friendly, easy-to-use, and intuitive. To use this technique, you need to: (1) learn about your users, (2) use that information to guide your design efforts, and (3) test out those design decisions with real users. The UCD technique is most effective when you use it iteratively throughout the development process, so that you can continue to refine and improve your design to enhance the way that users interact with your product. If you wait until the product is nearly finished to begin testing it with users, it may be too late to fix any major usability problems that turn up.

There are entire fields of study related to UCD and usability testing. This paper is not intended to cover these topics in depth, but instead to give you a brief overview of the basic concepts and provide an example of how the CAMEO development team has used UCD to improve their chemical emergency software products.

3 GETTING TO KNOW YOUR USERS

The first step in implementing the UCD technique is to get to know your users: the more you know about your users, the more you can work that into your design process. In general, you should make sure that you have a good idea of (1) what skills your users already have, (2) where the users will be working with your product, and (3) what tasks the users need to complete with your product. Here are some sample questions that you might want to know about your users in each of those areas:

- **Basic Information:** What jobs do they do? What educational backgrounds do they have? What languages do they speak? What computer skills do they have? What are their age ranges?
- **Work Environment:** What type of computers do they have? What software programs do they use frequently? Do they have software restrictions? Do they have an internet connection? If so, what speed is it and how reliable is it? Where are the computers located (e.g., in an office or in a hazmat truck)? Do they have access to a mouse, or will they be using a touchpad on a laptop? How large are their monitors? How loud is it in their work space?
- **Work Duties:** What tasks do they currently do that are similar to those that can be done with your product? What steps are involved in completing those tasks? Where do they get the source data for those tasks? What type of output do they need to produce for those tasks?

Knowing the answers to these questions (and others like them) can help you start thinking from the mindset of your users. There are two methods that you can use initially to start getting to know your users: (1) collect in-house knowledge, and (2) perform a domain analysis. You can also gather additional information about your users during usability tests (once you reach that stage in the development process).

* *Corresponding author address:* B. Connolly,
7600 Sand Point Way NE, Building 3, Seattle, WA
98115; e-mail: orr.cameo@noaa.gov.

3.1 Collecting In-House Knowledge

Particularly for long-standing products, you may find that your team already knows a lot about your users. You may have learned about your users while providing training, answering technical support questions, and receiving suggestions for improvement from the users. Try to collect this in-house knowledge in a single document that you can refer to once you start working on a new development project; you should try to capture both basic background information on the users and information about places in the existing design where the users struggle.

Although in-house knowledge can help you make UCD decisions, it is important not to overestimate how well you know your users. In-house knowledge about users can be incomplete and it can quickly become out-dated if you don't continue to work with the users to learn more about them.

3.2 Performing a Domain Analysis

If you don't know much about your users—or there have been major changes within your user group recently—you may want to implement a technique called a domain analysis. This technique involves doing one-on-one interviews with a number of users (typically 5-20 people). You want to ask broad, open-ended questions that encourage users to talk about themselves and how they do their jobs. Your questions should focus on topics that are of interest to your product, but the goal is to collect as much information as possible. After you've completed all of the interviews, you analyze the notes and pull out trends and interesting insights. By keeping the questions broad, you often end up finding out unexpected things about your users that have important consequences for your design. If possible, it is also a great idea to visit the locations where your users actually work so that you can see their typical working environment firsthand.

3.3 Primary vs. Secondary Audiences

Once you start considering the people who use your product, you will likely find that your users fall into several distinct groups (called audiences). For instance, the products in the CAMEO software suite are used by emergency responders, emergency planners, members of industry (such as industrial chemists), and academics (including professors and students at the high school and college level). Trying to keep all of those different types of users in mind during the design process can be complicated and, in some cases, may even dilute the usefulness of the UCD technique. Typically, it is best to focus one or two user groups (the primary audience for your product) when beginning to use the UCD technique. This allows you to more easily keep the concept of "your user" in mind during design decisions.

Making the initial concession to focus on a primary audience is important, but it doesn't mean that the secondary audiences are completely ignored during the design process. Once you start considering any group of users when making design decisions, you typically start creating products that are more straightforward to use. Many of the design decisions that you make for the betterment of the primary audience will also benefit your secondary audiences. Additionally, once you have gotten your product well-designed from the perspective of your primary audience, you can consider adding additional features for your secondary audiences in later versions—although you still need to keep your primary audience in mind to make sure that these new features don't cause confusion.

4 DESIGNING WITH YOUR USERS IN MIND

The second step in implementing the UCD technique is making design decisions with your users in mind. To achieve this, it can be helpful to single out one member of the design team to serve as the user advocate. This person should be very familiar with all of the information that you've collected about your primary audience, so that he or she can essentially act as a representative of your users at development meetings. As the development team discusses new projects or changes to existing products, the user advocate should consider the issues from the users' point of view and engage the development team in a discussion of any issues where the product development might be diverging from what the users need or want.

As mentioned in the previous section, this step is made easier when you focus on a primary audience. In the case of the CAMEO software suite, the primary audience is considered to be emergency responders—especially firefighters and hazmat response teams. Focusing on one group of users makes it easier for the user advocate to "channel" those users at development meetings and ask the team questions with the users in mind. (For example: Would firefighters use the new proposed feature? Will upgrading to a new software platform cause difficulties for fire stations with older computers?) If you try to consider too many user groups at one time, it can make it hard for a single user advocate to represent all of the users.

Ideally, the primary audience should represent the bulk of your users and/or the users with the most stringent design requirements. For CAMEO, the primary audience meets both criteria: emergency responders are one of the largest user groups, and their need for quick, accurate information during high-pressure situations in a variety of locations presents more challenging design requirements than those of some of the other user groups. Emergency responders also present an additional challenge for the CAMEO development team: infrequent use. Emergency response work is very unpredictable, so users are unlikely to use the CAMEO software suite daily. So, the CAMEO software needs to be as intuitive as possible, so that infrequent users can still get the information that they need in those high pressure response situations. One way to do that is to capitalize on the existing mental models of your users.

4.1 Mental Models

Mental models are the way that people think about things—that is, the way that people expect things to behave. If you can identify mental models that your users have about software (or the tasks that your software is used for), then you can leverage those mental models to make your software easier to use—especially for infrequent users. For example, if most of your users have a particular operating system, you can assume that daily use has made them familiar with the general features of that system. In essence, they have created a mental model of how things should work based on the system that they use frequently. People frequently try to apply existing mental models to new programs they are learning (for example, seeing if the cut-and-paste keyboard shortcuts work in the new program). The new program will be much easier for people to learn if some of the program's functionality already fits in with their existing mental models.

When you are creating new software designs, spend some time considering the programs and tools that your users already work with to see if you can identify any mental models that you can leverage in your design. For example: What keyboard shortcuts might users already be familiar with and want to reuse? Where do users expect to see help buttons or icons? Are users used to working with menus or with toolbars and palettes? What terminology are they familiar with in their day-to-day jobs and from other programs? What units (e.g., parts per million) do they use? What file types do the programs import/export? Thinking through these types of questions allows you to begin to get a sense of how the users expect programs to behave. If possible, try to make the users' transition to your program easier by using some familiar elements (e.g., button names and placement) in your design. There will be places where your design needs to diverge from the familiar; however, you want to make sure that these design elements are necessary—because the users will have to work harder to learn this new and different functionality.

5 USABILITY TESTING

The third step in implementing the UCD technique is testing your product with real users. Up to this point, you've spent some time researching your users and thinking about them while you made design decisions. This step lets you "ground truth" your design by watching how users interact with your program and finding out how well you succeeded in the first two steps. Performing usability tests with can help you identify a wide variety of issues with your program and help you prioritize the important ones based on how frequently they come up.

Typically, usability tests involve providing a few users with a series of tasks and then observing them as they try to complete the tasks. One technique that can be helpful is to ask the users to talk out loud during the tasks, so that you find out more about why they followed particular paths or why they became confused at some point. During the tasks, you try to interact with the users as little as possible (so that you don't compromise the test), but you can ask them specific questions after the exercise during a debrief session. Restraining yourself from interacting with the user during the test is probably one of the most difficult things about usability tests; it's hard to watch the user struggle to work with the program when you could just point them in the right direction. However, for usability tests to be successful you need to experience how the users would work with the program themselves (as if you weren't sitting there); you need to see the places where users struggle with the program in order to improve your design.

There are many resources available that describe how to run usability tests in detail, you should definitely consult some of these before trying to run a test yourself. You may also want to consult an expert (at least initially) to show you how to observe the tests accurately and how to analyze the results. Most of the usability test results are qualitative, since much of the data collected is from discussions and observations. However, you can also capture some quantitative results, such as the success rates on certain tasks or the average length of time to complete tasks.

Usability tests involve a lot of observation, so it is usually most effective to do them one at a time—that is, with one test subject and one observer. The test subjects should be representative users from your primary audience; generally, five subjects is a good number. (Five subjects is usually a large enough sample to turn up most of your usability issues.) Particularly during a major redesign, you may want to run several rounds of usability testing throughout the development process to test out new features.

Usability tests are also a great way to get to know your users better. You will learn a lot about the users by observing them during the tests (this is especially helpful if you are running the test at their typical work station). Additionally, before starting the usability test, you may also want to ask users to complete a brief survey to collect some of that basic background information about your users. For example, if your product is a GIS tool, you might ask users to rate how frequently they use online mapping tools and then list the specific tools that they use.

5.1 Paper Prototypes

For most usability tests, you will be testing out a functional version of your program. However, early on in the design process, you can use paper prototypes to perform usability tests. Paper prototypes are design mockups—typically, either hand-drawn sketches or simple mockups created with graphic design programs. You can save considerable time and effort for your design team by performing a usability test on a paper prototype—before any effort has gone into coding the design.

To run a usability test with a paper prototype, you need a page mockup for each screen the user might go to. Instead of clicking on buttons or filling in text boxes as they would on a real program, users point to the button on the paper mockup or tell you what they would type in a text box. The test moderator serves as the computer, showing the user new screens (the paper pages) based on the actions the user performed. This type of test is not as rich as a typical usability test (for instance, it won't show you how users mouseover terms or how they scroll around a page); however, these tests will typically point out major work flow design flaws and problems with terminology. By finding out these issues early, you've saved your team a lot of time and effort. Based on the results of the paper prototype usability test, you may be able to begin having your programmers work on a new design—or you may want to work through a few more paper prototypes to ensure that the design direction works well for the users.

6 AN EXAMPLE OF UCD

Whenever possible, the CAMEO development team uses the UCD technique as part of the software development process. The following is one example of how UCD was used during the creation of a new product: CAMEO Chemicals.

6.1 Background

Historically, the CAMEO software suite included:

- **CAMEOfm:** A database program for emergency responders and planners. This program included a chemical library with thousands of chemical datasheets with response information, a reactivity prediction tool (to see how chemicals might react when mixed), and several blank database modules (such as Facilities and Contacts) that users could use to fulfill data management requirements under the Emergency Planning and Community Right-to-Know Act (EPCRA).
- **ALOHA:** A modeling application that estimates threat zones associated with hazardous chemical releases, including toxic gas clouds, fires, and explosions.
- **MARPLOT:** A mapping program with that allows users to easily view and modify maps, as well as create their own map objects.

The programs are designed to interact with each other (for instance, MARPLOT provides an easy way to view CAMEO's facility locations or ALOHA's threat zones on a map); however, the programs can each be used separately as well.

A few years ago, based on user feedback, it became clear that many users would appreciate splitting the CAMEOfm program into two. Given that, the CAMEO development team decided to take the chemical library and reactivity prediction tool out of CAMEOfm and create a fourth program in the suite: CAMEO Chemicals. (For many emergency responders the chemical library and reactivity prediction tool were the primary part of CAMEOfm that they used; the other database modules—geared more towards planners—were used less often.)

6.2 UCD in the Development Process

It would have been simple to just separate the chemical library and reactivity prediction tool from CAMEOfm, and keep the user interface and functionality the same as they were before. However, the CAMEO development team took this as an opportunity to redesign these features in a manner that was more user friendly and that resolved known issues with the existing program. For instance, many users were unable to download CAMEOfm due to restrictions from their IT departments, so the team decided to make the new program into a website.

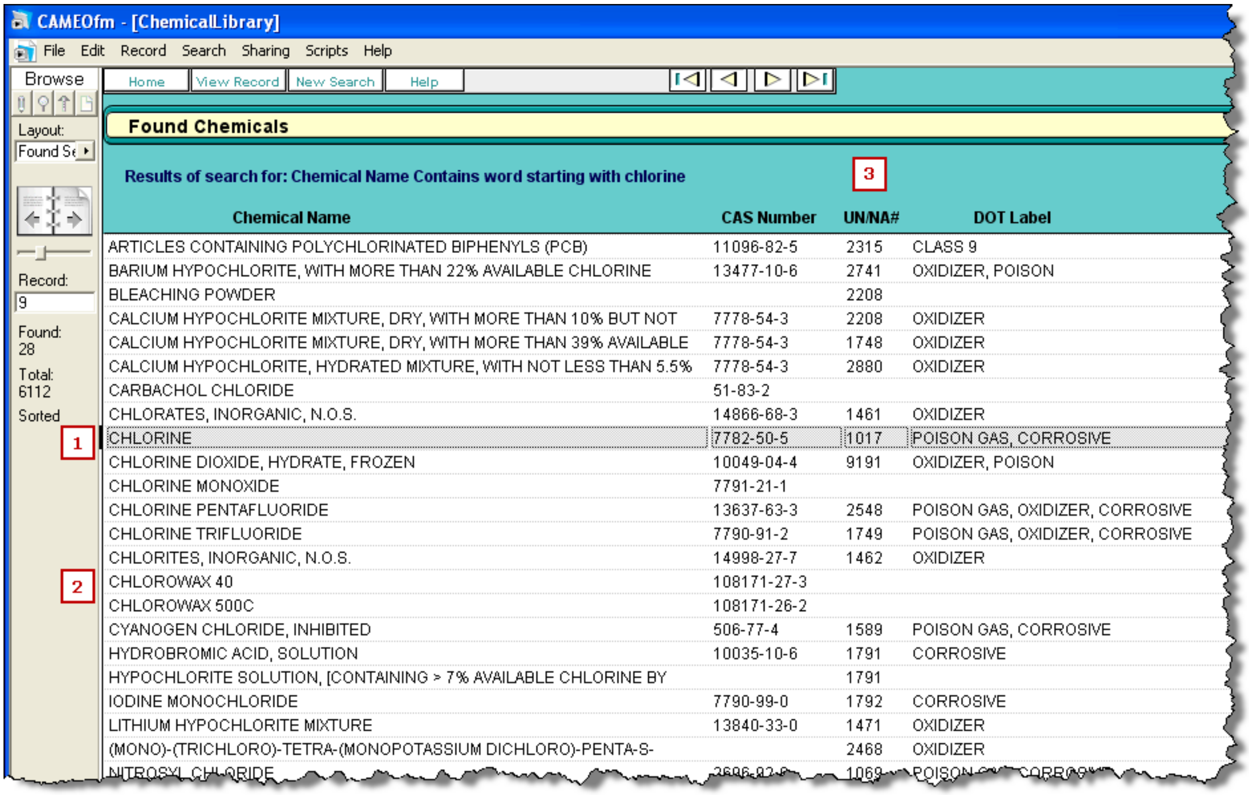
Once the team decided that CAMEO Chemicals was going to be a website, they began considering how to transition the existing content to this new architecture. While part of the team began discussing the technical issues of transitioning the content, another part of the team (the interface subgroup) began considering what the new interface should look like. (The interface subgroup included a web designer and two people with backgrounds in UCD and usability testing.) The interface subgroup used their knowledge of the users, the existing program issues, and general website guidelines to develop some initial mockups of the new CAMEO Chemicals website. These mockups were then discussed with the rest of the development team to ensure that they would work with the database backend.

After some refinements based on team input, the interface subgroup performed two paper prototype usability tests: one with a few in-house staff who were not involved with the project and another one with several members of a local fire station. These initial usability tests resulted in a multi-page list of small issues (such as places where terminology didn't match what the user expected or interface elements that needed to be more prominent), but there weren't any major usability issues that would require a re-design and more paper prototype tests. As such, the interface subgroup made some changes to the design based on the user feedback, and the programmers were able to begin implementing the design as a functioning website.

As the program development progressed, the interface subgroup continued to use the UCD technique to consider improvements to the design and functionality. Additionally, the interface subgroup was able to query a few users via email about specific issues as they continued working on the design. Once the programmers had completed a beta version of the website, the interface subgroup completed an additional round of usability testing at a local fire station. These usability tests were helpful because they identified some remaining issues, but they also verified the functions that users could work with easily.

6.3 UCD: Before and After

Figures 1 and 2 are before and after pictures of the search feature in CAMEOfm and CAMEO Chemicals, respectively. The figures highlight how UCD was used to make design decisions.



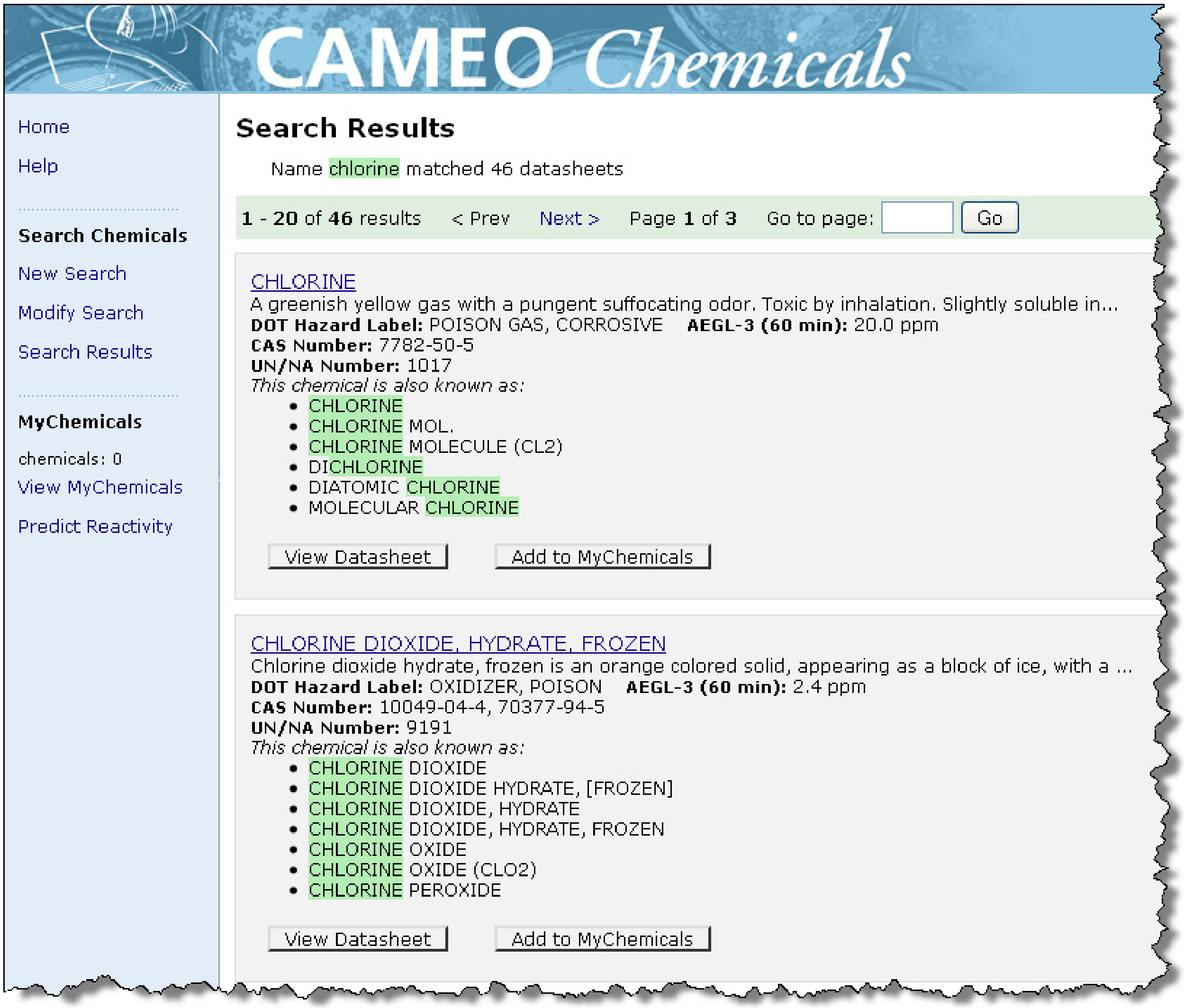
Found Chemicals

Results of search for: Chemical Name Contains word starting with chlorine

Chemical Name	CAS Number	UN/NA#	DOT Label
ARTICLES CONTAINING POLYCHLORINATED BIPHENYLS (PCB)	11096-82-5	2315	CLASS 9
BARIUM HYPOCHLORITE, WITH MORE THAN 22% AVAILABLE CHLORINE	13477-10-6	2741	OXIDIZER, POISON
BLEACHING POWDER		2208	
CALCIUM HYPOCHLORITE MIXTURE, DRY, WITH MORE THAN 10% BUT NOT	7778-54-3	2208	OXIDIZER
CALCIUM HYPOCHLORITE MIXTURE, DRY, WITH MORE THAN 39% AVAILABLE	7778-54-3	1748	OXIDIZER
CALCIUM HYPOCHLORITE, HYDRATED MIXTURE, WITH NOT LESS THAN 5.5%	7778-54-3	2880	OXIDIZER
CARBACHOL CHLORIDE	51-83-2		
CHLORATES, INORGANIC, N.O.S.	14866-68-3	1461	OXIDIZER
CHLORINE	7782-50-5	1017	POISON GAS, CORROSIVE
CHLORINE DIOXIDE, HYDRATE, FROZEN	10049-04-4	9191	OXIDIZER, POISON
CHLORINE MONOXIDE	7791-21-1		
CHLORINE PENTAFLUORIDE	13637-63-3	2548	POISON GAS, OXIDIZER, CORROSIVE
CHLORINE TRIFLUORIDE	7790-91-2	1749	POISON GAS, OXIDIZER, CORROSIVE
CHLORITES, INORGANIC, N.O.S.	14998-27-7	1462	OXIDIZER
CHLOROWAX 40	108171-27-3		
CHLOROWAX 500C	108171-26-2		
CYANOGEN CHLORIDE, INHIBITED	506-77-4	1589	POISON GAS, CORROSIVE
HYDROBROMIC ACID, SOLUTION	10035-10-6	1791	CORROSIVE
HYPOCHLORITE SOLUTION, [CONTAINING > 7% AVAILABLE CHLORINE BY		1791	
IODINE MONOCHLORIDE	7790-99-0	1792	CORROSIVE
LITHIUM HYPOCHLORITE MIXTURE	13840-33-0	1471	OXIDIZER
(MONO)-(TRICHLORO)-TETRA-(MONOPOTASSIUM DICHLORO)-PENTA-S-		2468	OXIDIZER
NITROSYL CHLORIDE	2696-92-6	1069	POISON GAS, CORROSIVE

Figure 1: A screenshot of the search results page from the old chemical library in CAMEOfm, where a search has been performed for records that have a *chemical name that contains a word starting with chlorine*. From interactions with the users, the development team had identified a number of priority issues to address; this figure highlights three such issues:

- *Issue 1:* Many searches resulted in long, alphabetical lists of matches, which often delayed users and caused irritation as they searched for the right chemical. Here, the user has searched for chlorine, and the most likely match appears as the ninth result in the list.
- *Issue 2:* The search function looks for matches in the official chemical name and any synonyms. This feature was intended to help the user (by being less restrictive about which chemical name field was being searched); however, this sometimes led to confusion when users reviewed the search results list and saw names (such as Chlorowax 40) that did not match their search term.
- *Issue 3:* The tabular search results screen only allowed a few details to be shown (Name, CAS Number, UN/NA Number, and DOT Label). Unfortunately, those details didn't usually provide enough information to allow users to choose quickly between similar chemical records.



Search Results

Name chlorine matched 46 datasheets

1 - 20 of 46 results Page 1 of 3

CHLORINE

A greenish yellow gas with a pungent suffocating odor. Toxic by inhalation. Slightly soluble in...

DOT Hazard Label: POISON GAS, CORROSIVE **AEGL-3 (60 min):** 20.0 ppm

CAS Number: 7782-50-5

UN/NA Number: 1017

This chemical is also known as:

- CHLORINE
- CHLORINE MOL.
- CHLORINE MOLECULE (CL2)
- DICHLORINE
- DIATOMIC CHLORINE
- MOLECULAR CHLORINE

CHLORINE DIOXIDE, HYDRATE, FROZEN

Chlorine dioxide hydrate, frozen is an orange colored solid, appearing as a block of ice, with a ...

DOT Hazard Label: OXIDIZER, POISON **AEGL-3 (60 min):** 2.4 ppm

CAS Number: 10049-04-4, 70377-94-5

UN/NA Number: 9191

This chemical is also known as:

- CHLORINE DIOXIDE
- CHLORINE DIOXIDE HYDRATE, [FROZEN]
- CHLORINE DIOXIDE, HYDRATE
- CHLORINE DIOXIDE, HYDRATE, FROZEN
- CHLORINE OXIDE
- CHLORINE OXIDE (CLO2)
- CHLORINE PEROXIDE

Figure 2: A screenshot of the search results page from the new implementation of the chemical library in CAMEO Chemicals, where a search has been performed for records that have *chlorine in their name*. This figure highlights how the development team addressed the three user issues presented in Figure 1:

- *Issue 1:* Previously, search results were alphabetized. When CAMEOfm was first developed, this worked well with the users' mental model. However, over the last few years, the users' mental model regarding search results has changed dramatically due to their use of internet search engines. Users now routinely expect the "best match" records to appear at the top of the search results list. In order to meet this demand, the CAMEO development team had to develop their own ranking system so that the "best match" records appeared at the top of the list. Not only did this change match up better with the users' mental model, it also helped users get to the information they needed more quickly—something that is very important for the primary audience (emergency responders).
- *Issue 2:* Previously, it wasn't always clear to users why records were appearing in the search results list. To resolve this issue, the CAMEO development team chose to highlight the search criteria at the top of the screen and have corresponding highlighting in the search results indicate exactly why the record matched. This highlighting technique is used by many internet search engines and search features in other programs, so it is something that users are already familiar with. Usability tests showed that users immediately picked up on the functionality of the highlighting (without requiring additional instructions).
- *Issue 3:* Previously, the tabular search results list offered limited space for displaying record details. In CAMEO Chemicals, the team switched to a list format with a larger abstract for each chemical record. The new abstracts included a brief excerpt from the general description as well as chemical identification numbers and levels of concern. In usability tests, it was shown that this new information helped users quickly identify the record that they wanted.

7 CONCLUSION

Here are the three key takeaways about implementing the UCD technique in your software design process:

- UCD can make your products more intuitive and easier to use.
- Implementing UCD means knowing your users, using that knowledge to guide your design, and then testing the design with users. Usability testing can be effective even with a small number of users (4-5 people).
- UCD works best when you start using it early in the design process and repeat the process iteratively during development.

There are lots of techniques for UCD and usability testing. The types you might use will depend a lot on your organization, the type of product that you are creating (in-house vs. external), and the budget for your project.