Giving Them What They Want:
Quick Prototyping on the HP3000

Lisa Burns Hartman
Hewlett-Packard
Corporate Offices
P. O. Box 10301
Palo Alto, CA 94303

If your experience is anything like ours, you have had mixed success with various methods of specifying systems. Specifying new systems is a difficult process, and can result in lengthy documents which are never read. Even for enhancements to existing systems, it can be difficult to express on the printed page what the new feature will do. Using diagrams of new screens can help, but it is not the same as logging on and interacting with a system. Users who receive systems which were designed only with written documents may be surprised and unhappy when the system arrives.

Prototyping

Our first answer to this problem was to build prototypes of new features. This was more successful than a written document for several reasons. Our users could see the screens online. This allowed them to try out the ergonomics of the screens: the video enhancements, the softkeys, the TABing, the placement and length of the fields, and the overall aesthetics of the screen itself. By interacting with a working prototype they could type in transactions, see data returned to them and clear up any confusion about how the new screen or feature was really going to work. This was infinitely more informative than reading a document.

However, building the prototypes was not cheap. We did not want to use a Fourth Generation Language because of the heavy interaction with our existing system for some of these prototypes. It was difficult to integrate prototypes written in a 4GL with existing programs in COBOL. So, since we used COBOL for the prototypes, it took 6 weeks or more to develop some of them. By that time, our investment in the design as we saw it was quite high. However, our prototype sessions proved that our idea of for a new feature did not always work in a real office environment. If users wanted significant changes to the initial design, it would be quite expensive to develop another prototype. It was very tempting to go ahead and install the code we had already developed and to ignore the users' input! We were not catching design problems or possible design improvements early enough in the process.

Prototyping using V/PLUS and ENTRY

Because of these experiences, we decided to try another approach. By using the FORMSPEC subsystem within V/PLUS, we designed screens for our new features. By linking these screens together and adding plausible data to be displayed to the user, we built a simulated prototype of the

new feature which could be run using the ENTRY subsystem of V/PLUS. This prototype included a narrative script where users are told to enter values in the various fields to set up a transaction. The entire setup for the prototype took our summer student, who was not familiar with FORMSPEC, less than two days. Not a single line of code needed to be written!

Advantages of ENTRY prototyping

When we sat down with users to show them the new screens, they were very pleased. As with traditional prototyping, they could see the screens online. They could review the placement of fields, which fields they wanted to see, the video enhancements to be used, and the overall appearance of the screen. They could enter transactions according to instructions in our narrative and work with TABing and length of fields. Again, this was much better than asking them to respond to a written document, and much more informative.

You may have already used ENTRY to allow users to try out screen designs. It is an excellent way to get feedback quickly. But blank screens have limitations. Without displaying data back to the users, we cannot meet the goal of simulating an interactive program. However, by linking several copies of the same screen together, and placing data in the fields in the second screen, we can simulate a program interacting with the user. The users can respond to the demo as though there was a program returning the data, and give us feedback on the way the function was designed.

How to set up an ENTRY prototype

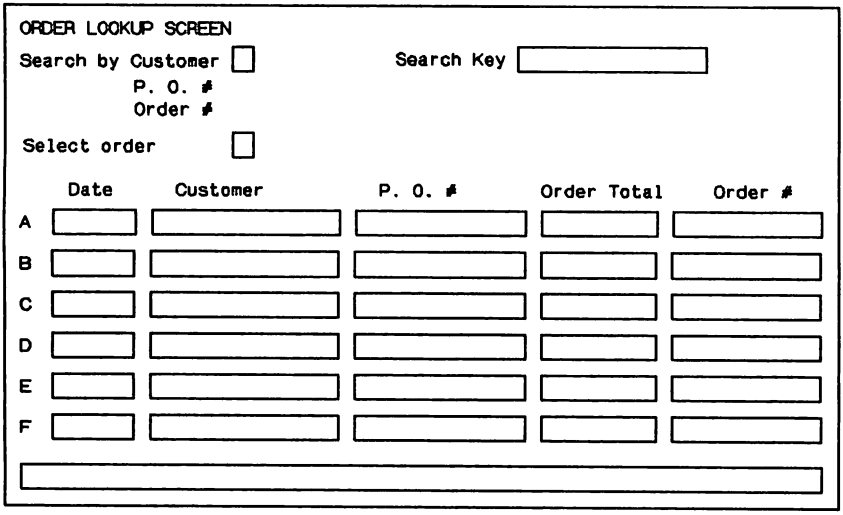
To illustrate how to set up such a simulation, let's take an example of an order inquiry screen. Suppose that we are going to design an inquiry screen to access an order database. The first step is to run FORMSPEC.PUB.SYS, and set up the new screen. Screen 1 shows the design of our first screen when brought up in ENTRY. This will be the first screen the users see when they run the prototype.

Now that we have our basic screen design, let's think about how our prototype will work. With this screen, the user will be able to query our database using three different keys: Customer, P. O. number, and Order number. How might a user expect this screen to operate? How will the searches work? Will partial keys be allowed? How will the orders be sorted when they do appear? Our prototype sessions with the users will answer these questions.

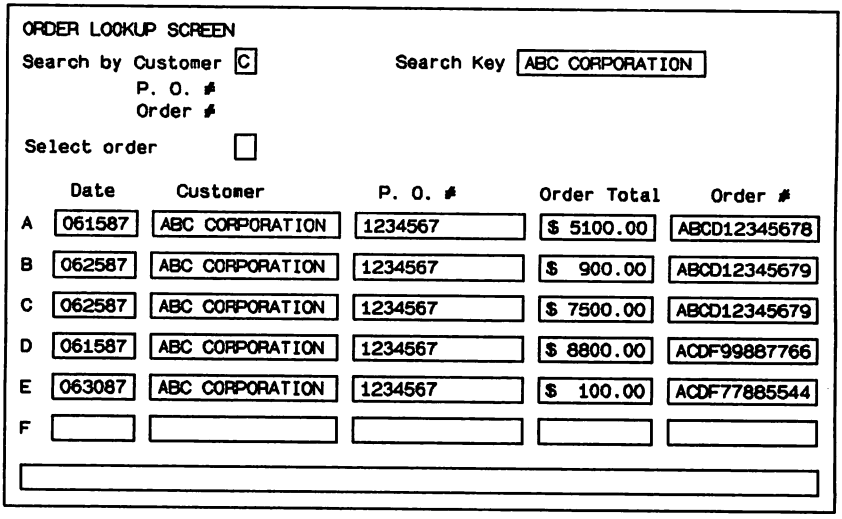
We need to set up demonstrations to answer each of our questions. First of all, let's set up an example for customer searching. First, we make a copy of the screen to a second screen name in FORMSPEC. Then, we link these two screens together using the NEXT FORM field. Then, working on the second screen, we use the Initial Values field in FORMSPEC to set up prototype data. I found that printing the sample screen and writing in plausible values for each field made the data entry of initial values in FORMSPEC much easier. The results of this data entry are shown in Screen 2.

In only a few minutes we have just written a small prototype! By running ENTRY.PUB.SYS and specifying the FORMSFILE name, we can run the prototype. The user first sees the blank basic screen (Screen 3a). Our narrative calls for a demonstration of the customer search feature. In our script, the user is asked to type a "C" for Customer search in the first field of the screen, and then to type the customer name "ABC CORPORATION" (Screen 3b). The user then hits <ENTER>, and the second screen we created is painted, with our sample data from the Initial Values fields in FORMSPEC (Screen 3c).

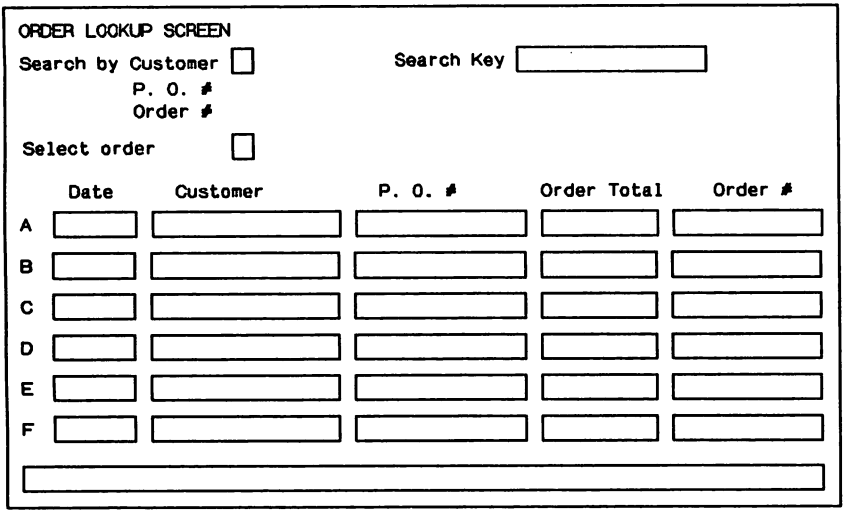
SCREEN 1


ORDER LOOKUP SCREEN
Search by Customer
P. O. #
Order #
Search Key
Select order
Date
Customer
P. O. #
Order Total
Order #
A
B
C
D
E
F

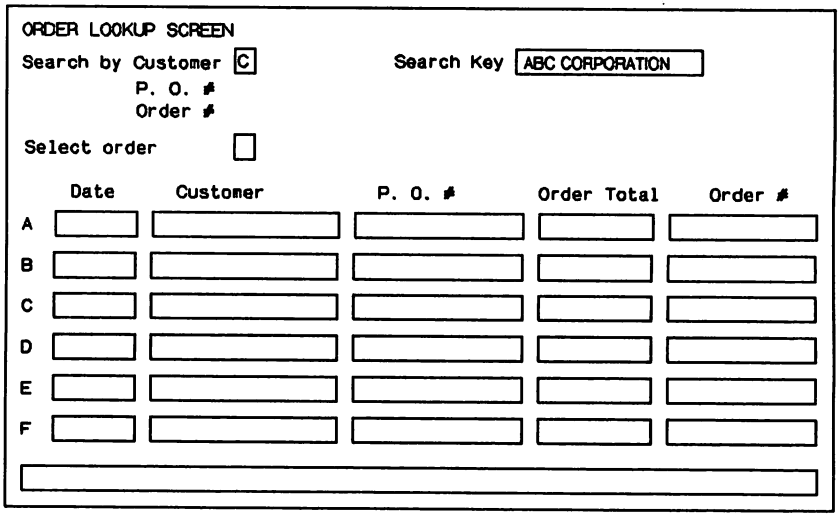
SCREEN 2


ORDER LOOKUP SCREEN
Search by Customer C
P. O. #
Order #
Search Key ABC CORPORATION
Select order
Date
Customer
P. O. #
Order Total
Order #
A 061587 ABC CORPORATION 1234567 $ 5100.00 ABCD12345678
B 062587 ABC CORPORATION 1234567 $ 900.00 ABCD12345679
C 062587 ABC CORPORATION 1234567 $ 7500.00 ABCD12345679
D 061587 ABC CORPORATION 1234567 $ 8800.00 ACDF99887766
E 063087 ABC CORPORATION 1234567 $ 100.00 ACDF77885544
F

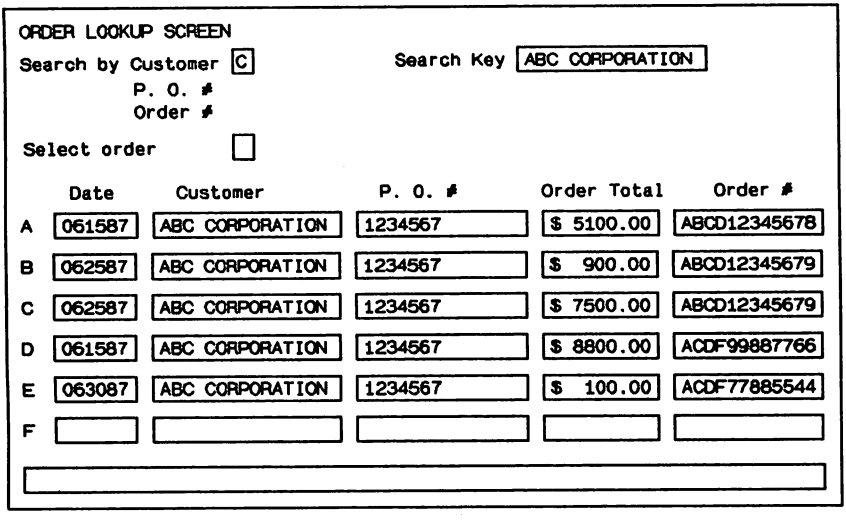
SCREEN 3A


ORDER LOOKUP SCREEN
Search by Customer
P. O. #
Order #
Search Key
Select order
Date
Customer
P. O. #
Order Total
Order #
A
B
C
D
E
F

SCREEN 3B


ORDER LOOKUP SCREEN
Search by Customer C
P. O. #
Order #
Search Key ABC CORPORATION
Select order
Date
Customer
P. O. #
Order Total
Order #
A
B
C
D
E
F

SCREEN 3C


ORDER LOOKUP SCREEN
Search by Customer C
P. O. #
Order #
Search Key ABC CORPORATION
Select order
Date
Customer
P. O. #
Order Total
Order #
A 061587 ABC CORPORATION 1234567 $ 5100.00 ABCD12345678
B 062587 ABC CORPORATION 1234567 $ 900.00 ABCD12345679
C 062587 ABC CORPORATION 1234567 $ 7500.00 ABCD12345679
D 061587 ABC CORPORATION 1234567 $ 8800.00 ACDF99887766
E 063087 ABC CORPORATION 1234567 $ 100.00 ACDF77885544
F

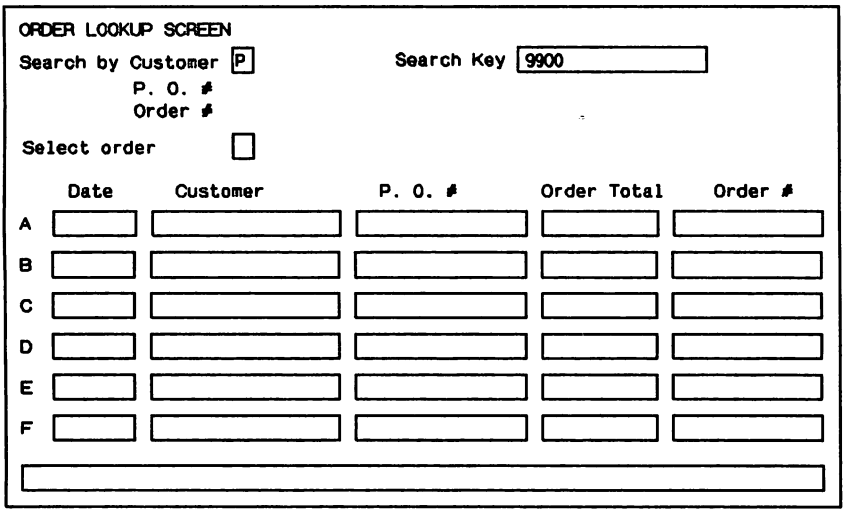
This example illustrates the basic concept we followed in creating our prototype: Show the basic screen, tell users what to enter in each field, then show the next screen filled with data corresponding to the entered data.

The same technique can be used to illustrate alternatives for implementing a particular program function. Screens 4a-d illustrate a data inquiry function which could display data in one of three different sort orders. Users viewing this prototype could choose the sort order which most closely fits their needs.

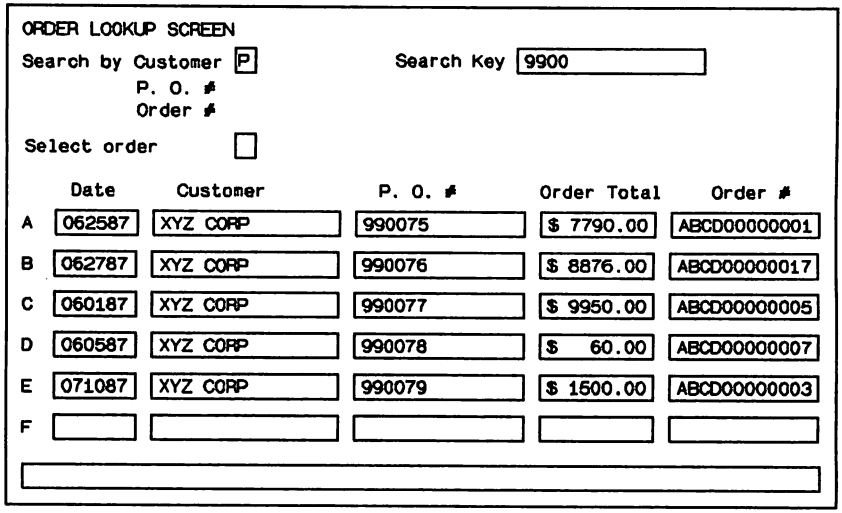
What else can an ENTRY prototype show our users? By making copies of the basic screen and then varying the copies, we can demonstrate the effect of different video enhancements. Screen 5 shows a variation of our basic screen with underline enhancements for some of the fields rather than half-inverse. ENTRY can also allow users to experiment with field placement. Screen 6 and Screen 7 show the same data elements arranged differently on each screen. By having users enter data from actual transactions into these screens, they can determine the best placement of the fields for ease of data entry. The placement of fields and the video enhancements used may seem trivial to a programmer, but to someone who stares at screens eight hours a day, these are very important issues.

By linking various scenarios together, we can set up a prototype session illustrating several alternatives for a new screen design. This session will yield very effective user feedback and help us resolve design issues before we write a single line of code.

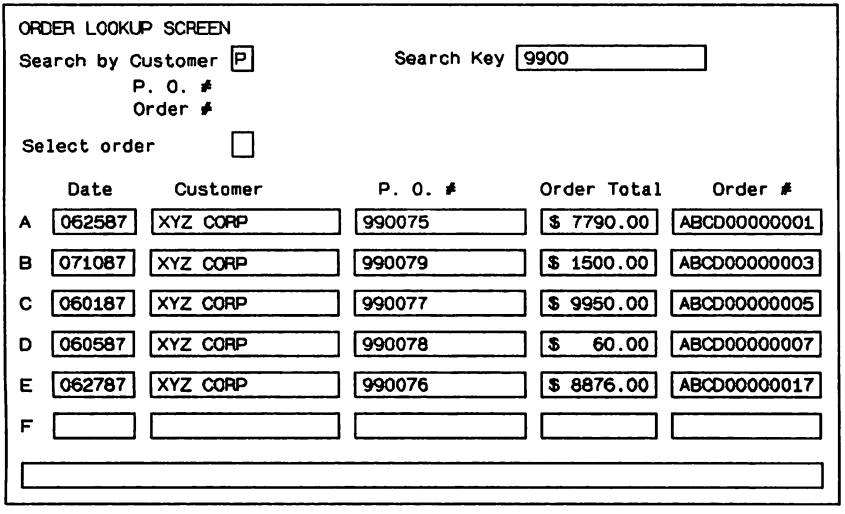
SCREEN 4A


ORDER LOOKUP SCREEN
Search by Customer P
P. O. #
Order #
Search Key 9900
Select order
Date
Customer
P. O. #
Order Total
Order #
A
B
C
D
E
F

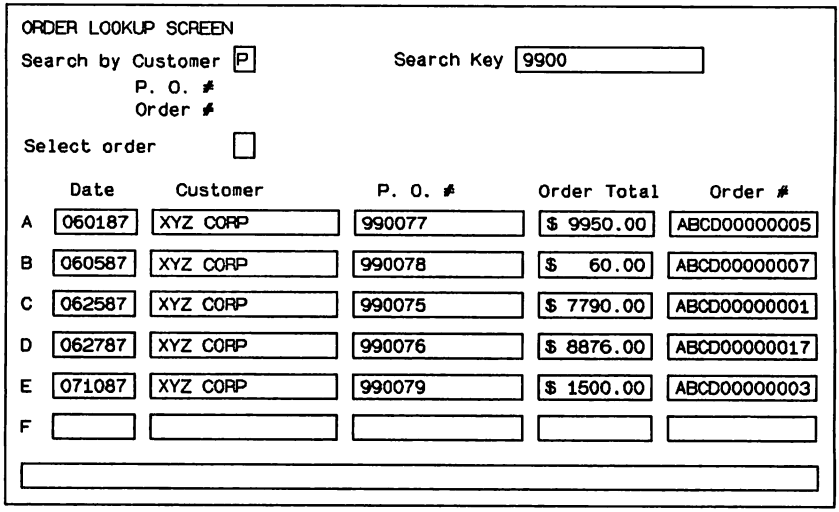
SCREEN 4B -- P. O. # sort



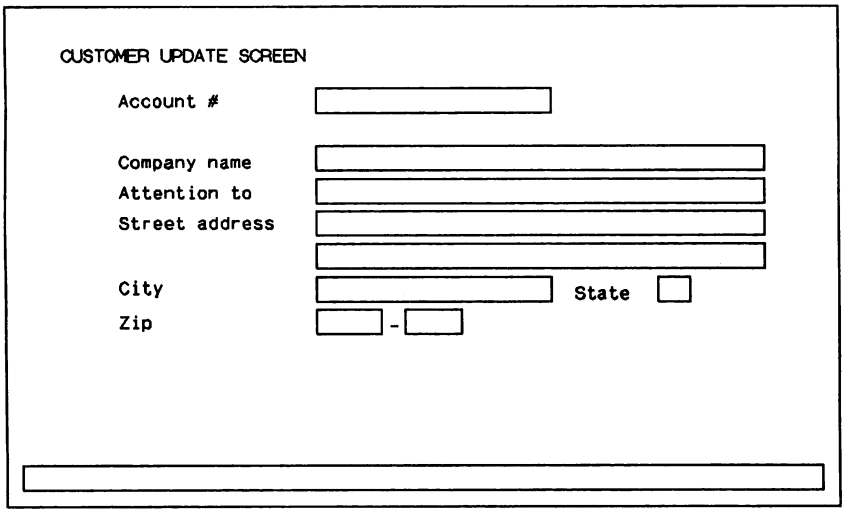
SCREEN 4C -- Order # sort



SCREEN 4D -- Date sort



SCREEN 7


CUSTOMER UPDATE SCREEN
Account #
Company name
Attention to
Street address
City
State
Zip
-

The prototype session

Writing the prototype is only half the battle for getting user input. By asking the right questions, we can elicit very important information from our users. The following are some tips we have found helpful when setting up an ENTRY prototype session.

- Schedule the prototype meeting at a time when users will have an hour or so to devote to this activity. We have found that our users are very excited to participate in the sessions, so getting them to take time to see the prototype is not usually a problem.

- Set up the prototype in a room where the users and the designers will not be distracted. This allows everyone to concentrate on the session.

- Set up several terminals, one for every 2-3 people. This allows each person to get hands-on experience with the screens.

- Give each user a copy of your narrative so that they understand the test data to be entered. Include instructions for running ENTRY.PUB.SYS and putting up the first screen. This will minimize confusion during the session and allow easy data entry according to the narrative.

Now it's time to begin the session. As each screen comes up, ask the users questions: Do you like the order of the fields? Do you like the video enhancements? Is the screen too busy? Are the field labels clear? and so on. Have them tab around the screen and enter sample transactions. Ask them if the fields are in an intuitive order, if seldom-entered fields are at the end, and if tabbing works the way they would like.

As you display each alternative design, ask which features they like the most: Does this sort order work best for you? Do you need a search on less than a full key? Is enough data displayed? Are error messages self explanatory? Does the system respond in a way which makes sense to you? Does it operate in a manner which fits the way you do business? You may need to ask leading questions in order to stimulate discussion.

You may wish to designate one of the designers as a scribe so that answers to these questions can be recorded. When the session is over, the scribe should summarize the results of the session, noting users' preferences and any changes to the screens which were suggested. As a final check, you may wish to run a second session with the combined

results in a new screen or screens. This will help ensure that you heard your users correctly.

Once your prototype sessions are completed, you should publish a document with printed screen layouts explaining your final external specification. Hopefully, this document will be far shorter than one generated without user input, since design alternatives have been eliminated and issues have been resolved. You can then proceed to construct the new screens or system, confident that you are creating something that your users will like and which will fit their needs.

Limitations and advantages

There are a few minor limitations of ENTRY prototyping. Since there is no application program running behind the screens, response time will not be accurate. Also, softkey functionality cannot be demonstrated for menu-driven applications. Finally, the cursor will always return to the first field on a given form. For applications doing cursor placement in a field somewhere else on the screen, this may confuse users.

The advantages of ENTRY prototyping outweigh the limitations for many applications. The technique allows users to interact with the system and provides most of the benefits that a prototype program provides. It is not necessary for designers to learn a new language. If their application is V/PLUS based they are already familiar with FORMSPEC. But by far the biggest advantage of ENTRY prototyping is the speed of development. I developed all of the screens for this article in under two hours. Modifications to screens can also be made very quickly.

Conclusion

Using FORMSPEC and ENTRY to set up prototypes can be very valuable for application teams. A day or two invested up front can save weeks and months of redesign later in the development cycle. ENTRY prototyping lets the user interact with the proposed system hands-on, on-line. By showing users exactly what the system will do and letting them see and touch a prototype, developers can be confident that the product they will deliver is what the user wants. Our experience with ENTRY prototyping has been extremely positive. We plan to expand its use for new enhancements to our systems.

A version of this article originally appeared in the October, 1987 of HP Professional magazine.

Prototyping on the HP3000
2068-20