

Getting a FRAME Entry Application Built in a Research Environment

Charlotte Corelle, Kaiser Permanente Center for Health Research, Portland, Oregon

ABSTRACT

The author tells the story of a menu-driven system built at the Center for Health Research to let researchers use information in the automated pharmacy database of a large health maintenance organization.

INTRODUCTION

I work as an analyst in the Research Assistants department of the Kaiser Permanente Center for Health Research (CHR). In 1994 I was part of a team at the CHR that designed, coordinated, and implemented the first phase of a medication surveillance reporting and data-building application (GAME). The job was technically complex, but it was even more challenging as an organizational activity conducted at a particular time in a particular work culture. What we did was translate a research investigator's very specific needs into a system that could be built at the Center for Health Research. The resulting application used new hardware and software. It required senior management support and cooperation both across and within the technical staffs of three CHR departments: Computer Operations, Programming, and Research Assistants. It also involved hiring outside consultants.

When we started building GAME, I was already a proficient SAS® user but had not done any previous applications programming. My technical skills grew a lot during the experience and so did my understanding of non-technical challenges that can influence systems development within an organization. I want to share some of what I learned, especially what I found most interesting. I will provide an overview of the application itself and give you my perspective of the context and circumstances involved in building it. If there is a system in your future, I hope you will find something in our experience to help you in yours.

BACKGROUND

The CHR has a staff of approximately 200, including twenty investigators with a variety of disciplines—including biostatistics, dentistry, economics, epidemiology, gerontology, medical care organization, medicine,

nursing, nutrition, pharmacy, psychology, social psychology, and sociology. The investigators pursue a program of research in two broad areas. One program covers health services research and social and economic studies; the other focuses on epidemiology and disease prevention. The CHR uses Kaiser Permanente, Northwest Region (KPNW) as a research laboratory. KPNW is a health maintenance organization (HMO) with 378,000 members located in the Portland, Oregon and Vancouver, Washington metropolitan areas. With few exceptions, its members receive all of their health care from the HMO. The CHR staff uses the HMO's automated databases as major data resources.

The HMO's databases include an outpatient pharmacy system, an automated hospital admission and discharge database, a tumor registry, and a KPNW membership eligibility database, among others. The HMO assigns to each member a unique identifier and uses it to label all data for services a member receives. The unique member identifier makes these data especially valuable to the CHR because it allows us to link data from the various databases. For example, we can select all outpatient pharmacy dispensing records for drugs a researcher wants to study. Then, using the unique member identifier from the dispensing record, we can map to other KPNW databases to get demographic, morbidity, and health care data about the drug users.

We developed the application, which came to be known as GAME, with funding from the CHR's participation in a cooperative agreement (#FD-U-000739-03) with the Food and Drug Administration to conduct studies on adverse effects of prescription medicines (ADE). A CHR investigator made the initial request for a system in October of 1991, near the beginning of the three-year funding cycle for the agreement. At that time the CHR computer environment consisted mainly of VT320 terminals linked by Ethernet to a local VAX cluster, which was in turn remotely linked to other VAX machines in the KPNW offices. Transactional data for the HMO were routinely downloaded by KPNW computer operations staff to databases on the remote VAX machines and stored in S1032, a relational database software marketed by Praxis International.

RESPONDING TO A QUERY BEFORE GAME

One of the earliest queries we handled occurred when the Agency received an inquiry concerning the possibility of an association between Sudden Infant Death Syndrome and the use of fluoride drops, a medication frequently prescribed to prevent tooth decay. That query was an example of the type of request we needed the capacity to quickly address. As the first step in responding to the inquiry, FDA scientists asked the CHR to identify all infant users of fluoride drops within the HMO. Because we handled the fluoride drops request before developing GAME, it illustrates the "old way" of initially identifying users and providing reports about them. The "old way" meant an analyst would search the HMO's S1032 drug product data set containing 10,000+ drug product description records and attempt to isolate a set of drug records with the appropriate product number values for mapping to the very large KPNW S1032 dispensing data sets.

The analyst would then print the list of product number records and give it to a CHR investigator with appropriate pharmacy background. If the investigator and analyst had time to look at the list when the analyst delivered it, the two would often discuss it, sometimes while the investigator referred to personal, hard copy records of drug products in the HMO data set. The reviewing investigator might approve the list "as is" or eliminate some drug records from the list and/or add others to it. Occasionally the investigator might need to consult with a working pharmacist in the HMO in order to respond to the list, a situation which could cause a delay.

Usually there would be some lag between the time the analyst printed the drug list for the investigator and the time when the investigator returned the approved or revised list to the analyst. The analyst might then take the revised list and, using the investigator's rationale for her revisions, search for records the investigator may have overlooked. If the search yielded additional product number records, the analyst would then print another list and again bring it to the investigator for approval. Until the analyst had the correct set of product number records, she could not begin searching the dispensing data set(s) to identify drug users, so immediate initial response to the query was usually impossible.

SYSTEM CHALLENGES

In the most general terms, the charge to the ADE team

was to develop application software designed to facilitate access to automated pharmacy data. The CHR investigator's initial request included mock-ups of several reports that he wanted to be able to generate quickly and accurately in response to ad hoc queries from the FDA. Considering the nature of the CHR's agreement with the FDA, the request was reasonable.

In 1996, within business environments, building a menu-driven system for executing the queries and producing the reports might seem an obvious rather than an innovative solution to the investigator's request. Remember, however, that the investigator's request was made in 1991. At that time acceleration in the emergence of new office technology had begun to create a kind of mania, especially in offices that were already fully automated. Many people in those environments were (and many still are) striving to appear less daunted by the emerging technologies than they actually felt. Probably at the extremes of reactions were, on one end, former techno-experts feeling suddenly clueless and, at the other end, techno-phobes feeling more scared than ever that they had missed the boat—this time for good. Optimists in some environments went overboard trying keep up with the technology while futurists gleefully predicted its effects on daily work life. In 1991 we were not accustomed to a demand for information services out of synch with the match of hardware, software, and skills in our possession at work. That situation feels more normal now. Back then, what felt normal for programmers was simply to have more work than time to do it in. Suddenly technical staff were asked to do work we could not do—not only because there was too much of it, but also because we lacked the right combination of knowledge and machines to do it with.

Also remember that the work setting of the request is not a business. The CHR conducts scientific research. Calm consideration, detail, thoroughness, and repetition are hallmarks of the culture of research science. Business managers value quick reaction. Scientists have high respect for patience, caution, deliberation. Scientific research is the business of carefully formulating and testing one hypothesis after another. This was the setting in which we would translate the investigator's request into a working application.

Although the culture of science pervades the CHR environment, technology has added an interesting flavor to it as well. The Director of the CHR decided around 1980 that the CHR should become totally automated and decreed that each worker should have direct access to E-mail, word processing, and data. By 1985 it was

so, and all employees had terminals and communicated via E-mail, whether they wanted to or not. Investigators made little use of the 1980's technology for accessing data directly. Research assistants acquired the necessary programming expertise and did it for them. The CHR became dependent upon technical staff in order to carry out its research mission.

Many cultural issues at the CHR collided around the investigator's late 1991 request. At that time there was increased pressure to maintain status as a work setting on the cutting edge of office automation technology. Unfortunately it was also unclear then just which direction to head in order to proceed with the wave of change: should we completely ditch the few MAC products we were experimenting with? What about the VAX workstations we had thought might provide all or part of our move to a "new" technology? We talked about UNIX machines, AXP's, Windows NT, and mostly about Microsoft Windows for PCs. Some who had already struggled to accommodate themselves to the office automation technology resisted the change. Individuals and departments had varying degrees of power to affect the direction and speed of the change.

For better or worse, the investigator's request provided an excellent candidate to develop as a major, menu-driven system which would apply the CHR's inevitable, though as yet undetermined, cutting-edge technology. In other words, the system became an early test case and battleground in the struggle to determine the CHR's future technology. The investigator's request for a system also contained an expectation that inherently added to the culture clash of the time at the CHR: he wanted a system for responding right away to ad hoc queries from the FDA. The "right away" was out of context with scientific research culture and foreign to the CHR. Until the CHR entered into the FDA agreement, staff had little or no previous experience in responding rapidly to queries from an outside funding agency. An obligation to do so simply had never been part of a grant project at the CHR.

It took us a long time to sort out how we could implement a system to simultaneously achieve quick turnaround time, research quality accuracy, and report presentability.

Until now I have described cultural factors at the CHR that added to the difficulty in getting the system built. There were, of course, other, non-cultural factors that were also difficult. These factors seemed more concrete and tangible, and though they were challenging to overcome, they were also more fun. For one thing,

working with the outpatient pharmacy data of any large HMO would be challenging. At KPNW one record exists for each unique pharmacy dispensing event since 1986 for each patient. When we use these data at the CHR, major challenges arise from the size and location of the pharmacy database and from its structure. The same is true about the hospitalization data that we use to search for adverse effects of prescribed drugs: the data content, size, and structure all present challenges to miners of information.

The KPNW dispensing records for 1986-1994 consume more than 4.6 million blocks of disk space. Altogether there are more than twenty-seven million records for those years. Each year's records are in separate data sets. Every record is 176 columns long and contains twenty-four variables. Among the variables on the dispensing sets are the drug product number and the unique member identifier fields. Only by searching the drug product data set can a researcher determine the particular product number values that apply to the drugs of interest.

The appropriate product number values can be hard to find because drug data themselves are complex. Drug chemistry is complicated. So are the ways and purposes for which doctors prescribe them, drug companies package them, and pharmacists classify them. A researcher first must select a complete and correct set of drug product number records and only then can map to appropriate dispensing records for accurately identifying users of a drug of interest.

Also, the data are clean enough for management reports, but often not quite clean enough for research standards. Most important, because of network issues the CHR does not control, remotely searching the database in S1032 can be a very slow process. Remotely doing the necessary data manipulating in S1032 at the CHR is hopeless; a researcher could grow old and die waiting for the reports to come off the printer.

RESPONDING TO A QUERY NOW

The initial version of GAME was ready at the end of 1994. Now that GAME exists, in most cases we are able to make an immediate initial response to the FDA in a standard report format. I'll describe what happens now when we receive an FDA query related to a possibility that users of a certain medicine may be experiencing adverse medical effects.

Let's say the query concerns some tricyclic antidepressant drugs sometimes also prescribed to prevent chil-

dren from bed-wetting and that the FDA suspects neurological damage may have occurred among children taking these medicines. Today an investigator or an analyst or both would sit down at either's workspace. If they sit together, the analyst is usually the "mouse driver," so she clicks the GAME icon that sits on her Windows desktop. This fires up a SAS FRAME Entry application that asks for her username and the name of the VMS project directory for storing the query output and billing the query processing (Figure 1). There is also a field she can use to write a text string to tell what the query is about. The information from this screen identifies the query session and appears as a footnote in all reports produced from the query session. The information here is also automatically added to a SAS data set that tracks who, when, and how researchers use GAME. When the "driver" finishes filling in the blanks, a second screen appears, the Welcome to GAME screen (Figure 2).

Figure 1

Figure 2

The Welcome screen has four big push buttons, one for each module of the application. The "driver" can choose

whichever one she wants to start with and click. The EFFECTS module has not yet been developed. When the module is built, she would use it to specify the type of adverse effects suspected, in this case diagnoses related to neurological damage. Currently, if she were to click on EFFECTS she would get the screen below (Figure 3). Instead, she chooses the DRUG PRODUCTS module, clicks, and this screen appears (Figure 4). From here, when she clicks on the SELECT PRODUCTS button, a highly interactive screen appears (Figure 5). Together she and the investigator use this screen to get instant drug product information and feedback that allows them to correctly specify the product number values of the exact drugs they want to study. If they wish, they can go back and forth between the two screens of this module to create sub-groups within the drugs they will study. When the analyst and investigator are satisfied that they have selected the appropriate drug product number values and groupings, they finalize their choices by clicking on the OK button on the first screen of the DRUG PRODUCTS module.

Figure 3

Figure 4

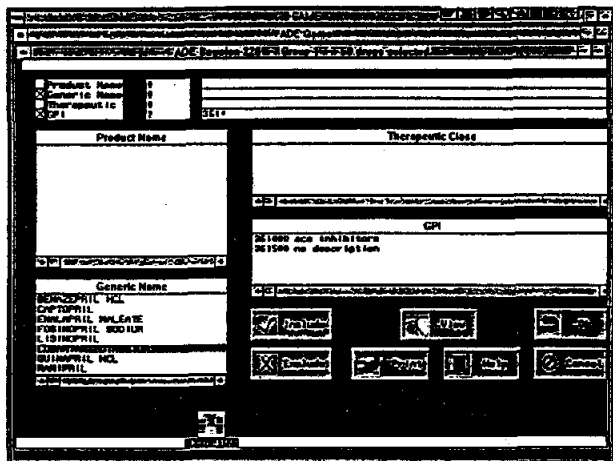


Figure 5

Once again they are looking at the Welcome screen. Now they click on the POPULATION module and a new screen appears (Figure 6). They use this one to enter the age and sex parameters of the population of HMO members they want to study. On this screen they also specify the years of membership and pharmacy records to study. In the process of specifying age parameters, they select the date during each study year to use for calculating age and the age categories to be used in query reports. When they are satisfied that they have appropriately specified the study parameters of HMO members for year, age, and sex, they click on the OK button.

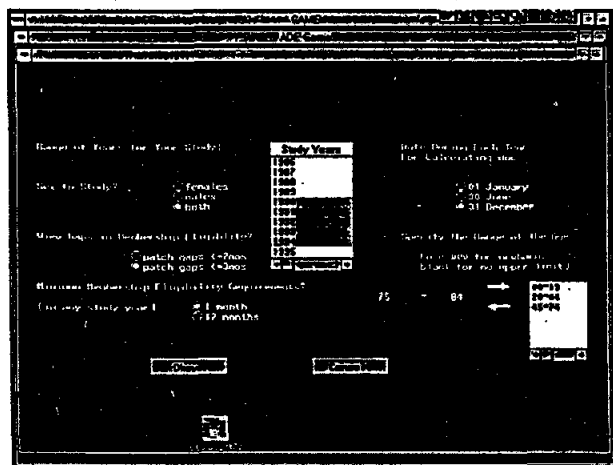


Figure 6

Again they are looking at the Welcome screen. Now the analyst clicks on the REPORTS module. The screen that appears (Figure 7) contains a list of all the types of reports and data sets that the current version of GAME can generate. The analyst and investigator decide which report and data set types they want, and the "driver" highlights their choices. Next the analyst clicks on the OK button, and the DOS screen appears and asks for a

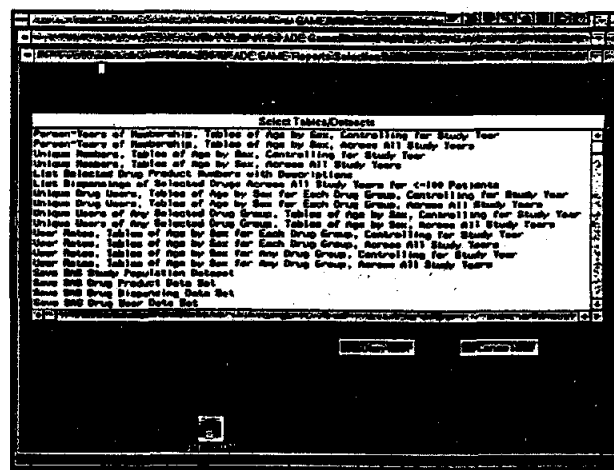
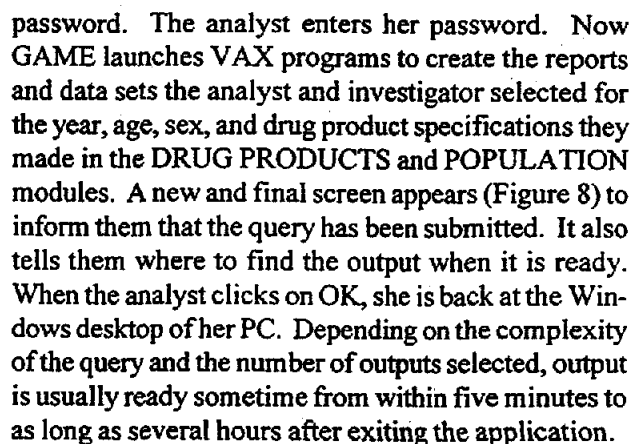


Figure 7

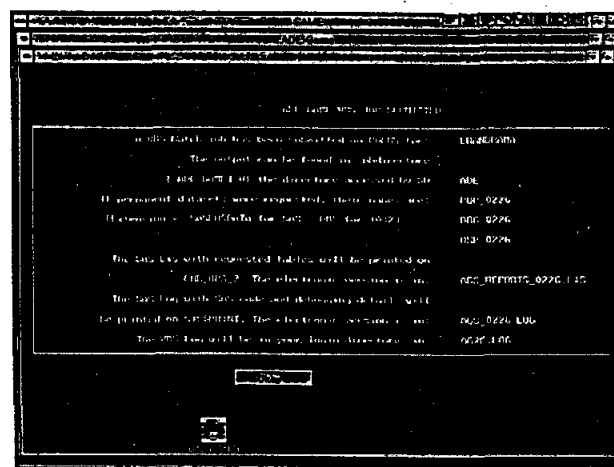


Figure 8

SYSTEM DESIGN AND DEVELOPMENT

The design of GAME is modular for two reasons. The first reason was that we could conceptualize as distinct activities involving distinct data sources, the processes that would be required to obtain and shape the

data for the overall set of reports the investigator wanted. The second reason for the system's modularity was so we would be able to stage development in a way that would maximize productivity. The distinct modules allowed us to schedule work so that the first finished module would provide immediate value, even though the whole application remained incomplete. They gave us flexibility to match available programming resources and skill levels to appropriate, available tasks. When resources were plentiful, we could choose to deploy them simultaneously on the same or different modules. When resources were scarce, they could still be applied to appropriate work. By the time we had settled on SAS/AF® as a development tool, two years had passed and it was crucial to get something of value developed as quickly as possible.

Back in 1991 we had started to work on the investigator's request by putting a lot of time and energy into a classic system design effort. The Programming and RA staffs had recently received training in the Yourdon method and team members eagerly applied the methodology. We drew a context diagram for our system, entity relationship diagrams, data flow diagrams, leveled data flow diagrams, and even an extensive data dictionary. If we had succeeded in pinning down an absolutely final version of the desired system's inputs and outputs, it is possible that the Yourdon method would have produced an excellent system eventually. After many meetings and several months of work, we still did not have final specifications. The possible permutations seemed endless. We stalled. Team resources were assigned other tasks. Eventually that initial design phase just fizzled.

Somehow we picked up speed again in 1992 and tried to make progress by deciding on an appropriate development tool. This was a can of worms because there were so many opinions and so many possibilities of what the appropriate development tool might be, given the certain movement away from our almost exclusively VT320 terminal and VAX environment and the uncertain direction of this movement. The Programming Department favored Microsoft Access. They were motivated by that product's GUI interface and relational database capabilities and by its immediate availability on the market. I argued for choosing either S1032 or SAS as a development tool because the HMO source data were stored in S1032 and would need to be manipulated in SAS. We had a working interface on the CHR VAX for translating between the two softwares. We also already had SAS/AF licensed for a few PC's and VAX workstations and knew that the SAS Insti-

tute would soon be releasing its FRAME Entry product for Windows which would be loaded with GUI features. I did not see the need to introduce another type of software to a system that was already an obvious challenge to build. My argument swayed no one, and Microsoft Access was picked for developing the system.

We spent a lot of time in 1992 learning to understand the HMO's drug product data set and cleaning it up in preparation for building our the system. During the year a few staff in the Programming Department had their VT320 terminals replaced with 486 PC machines with Microsoft Windows. When we were just about ready to do some coding on the application, we got news that the HMO would begin to use a new drug product data set in 1993. There was no way to find out anything about the new data set until it was in place. Once again progress on the system stalled.

Life went on. The new product information data source arrived and we pondered how to assimilate its changes to a system we had envisioned referencing the old data source. As months passed, it became clear even to me that PC Windows would be the choice for maintaining cutting edge technology at the CHR. My colleague in Programming experimented with Microsoft Access and drew prototype screens for a DRUG PRODUCT module. At SUGI in 1993 I discovered that S1032 was never a serious development tool candidate for GAME; it couldn't be used on a PC. S1032 was incompatible with the technology that offered the best selection of new bells and whistles. The SAS Institute released its Windows product for FRAME Entry.

Being pragmatic about leaving the familiar world of the VT320 and VAX environment and also wanting to see GAME built before the funding agreement ended, I preferred to enter the Windows world with something to hang onto. For me that meant choosing to learn v6.08 of SAS/AF for Windows as the GAME development tool, even though I still had a VT320 on my desk, the CHR had not yet installed the new FRAME Entry tool, and I had never done applications programming before. I made a case to fly to Irvine along with another analyst from the RA department to attend the first public SAS/AF for Windows class offered by the SAS Institute. A couple of weeks before I departed for the November 1993 class, the Computer Operations people hauled my VT320 away and plopped a new 486 PC with Windows in its place. The new order had arrived.

In Irvine I found the SAS/AF class difficult, mostly because I had never used SAS Screen Control Language

(SCL). Using the FRAME Entry product as a screen drawing tool was fun, but SCL was another beast entirely. I got from the class a strong appreciation for the power of SCL when combined with the SAS/AF objects a developer can place on a screen with ease and speed. At the point of my greatest despair while sitting in the class wondering what I had got myself into, the idea for the Welcome to GAME screen popped into my head and I sketched it on a piece of paper. That image provided an organizing principle for the system modules we had conceptualized. It gave me a vision of how to turn the investigator's request into reality.

As soon as I got back to Portland, I built the very same Welcome to GAME screen that still anchors the application. I also built an extremely primitive version of the POPULATION module of GAME. It took up three screens in splendid variety of fonts and colors. Pretending more confidence than I felt, I showed the screens to the principal investigator and project team and broached the subject that perhaps we should reconsider using SAS/AF to develop the now almost mythic GAME. I did my best to sell the idea that we could get the application built before the funds ran out if we used SAS/AF as the development tool. The Programming department was not convinced. The Computer Operations staff (whose cooperation was essential if GAME was ever to rise from the ashes and soar into existence) was non-committal. The principal investigator was mildly pleased. 1993 rolled into 1994.

In January of 1994 I checked in with my colleague in the Programming Department about how she was doing using Microsoft Access on the DRUG PRODUCT module and shared how much trouble I was having getting beyond what I had shown the team after returning from the class. I had succeeded in forming a fairly clear picture of what our system needed to do to work and how it needed to look, but I was at a loss about the nuts and bolts of building it myself. My colleague acknowledged she was also moving slowly in Microsoft Access and not having much fun either. We commiserated and decided that we each might go faster and enjoy the process of working with the new technology more if we worked together and committed ourselves to SAS/AF as the GAME development tool. This decision enabled us to make good progress coding the application. It also helped that the end of the first three-year funding period for the agreement was coming up at the end of May. I got permission to hire a consultant to help us deliver our system for improving access to the pharmacy data before the funding ended. At SUGI that year I connected with consultants. Those connec-

tions resulted in hiring Jessica Yuan through SAS Consulting Services and also Geoff Wheatley from the Portland SAS User Group, who had just started his own consulting practice. It also helped that we prepared for the arrival of the consultants by carefully outlining the work we expected them to do for the dollars we could spend. Our early work in Microsoft Access and SAS/AF proved useful in communicating our design to the consultants. Another trick my colleague and I used to get the application built in 1994 was scheduling presentations about GAME at various intervals and to several audiences before the funding period was over. Just before the last presentation, I begged my colleague to improve my early version of the POPULATION module. She agreed to do it, and a working system to respond to the FDA was in place by the end of 1994.

CONCLUSION

The application we built is innovative, even for 1996 and perhaps especially within a health research center. It provides a friendly Windows type interface for obtaining a lot of information directly from the HMO's very large pharmacy database. We used new and challenging technology to build it and in doing so expanded the CHR's technical capacity. Its construction required management and staff cooperation across and within departments. To obtain this support and cooperation, we built the application with features that would address needs of other projects at the CHR just as well as they fit needs of the funding project. We added utilities that would appeal to staff in both the Programming and Research Assistants Departments and also met a CHR goal to improve investigators' direct access to data. It surprised me that GAME took such a long time to build, but I am more comfortable with the delay and understand it better now that we have used the system for more than a year. It also surprises me that staff on few projects besides the funding project make use of GAME. We have recently completed user documentation and hope that will inspire researchers to use it. Within the last year we have added new outputs that are available from GAME. At this time we have done no further work on the EFFECTS module.

Building GAME sometimes felt like a normal pregnancy. It was a personal experience of growth and wonder, but I was often impatient with its on-going discomforts. Now that it is over, and the baby is healthy, and I have recovered, I would say our effort was worth the suffering.

ACKNOWLEDGMENTS

SAS and SAS/AF are registered trademarks of SAS Institute Inc. in the U.S.A. and other countries.

*Indicates U.S.A. registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

AUTHOR CONTACT

Charlotte Corelle
Kaiser Permanente
Center for Health Research
3800 N. Kaiser Center Drive
Portland, Oregon 97227
(503)335-6740
FAX (503)335-2424
corellech@chr.mts.kpnw.org