10/12-27-95 JSD

SLAC-PUB-95-7062
December 1995
CONF-9511152--2

# The PEP-II/*BABAR* Project-Wide Database Using World Wide Web and Oracle*CASE

Andrea Chan, George Crane, Ian MacGregor, Steven Meyer
*Stanford Linear Accelerator Center*
*Stanford University, Stanford, CA 94309*

The PEP-II/*BABAR* Project Database is a tool for monitoring the technical and documentation aspects of the accelerator and detector construction. It holds the PEP-II/*BABAR* design specifications, fabrication and installation data in one integrated system. Key pieces of the database include the machine parameter list, components fabrication and calibration data, survey and alignment data, property control, CAD drawings, publications and documentation. This central Oracle database on a UNIX server is built using Oracle*Case tools. Users at the collaborating laboratories mainly access the data using World Wide Web (WWW). The Project Database is being extended to link to legacy databases required for the operations phase.

## OUR ENVIRONMENT

The *B* Factory at the Stanford Linear Accelerator Center (SLAC) is a project built to study the physics of matter and anti-matter. It consists of two accelerator storage rings (PEP-II) and a detector (*BABAR*)–a project of approximately $250 million with collaboration by many labs worldwide.

We faced the challenge of integrating both administrative and technical data into one CASE enterprise design. The goal, defined at the project's inception in late 1992, was to use a central database as a tool for the collaborating labs to:

1. track quality assurance during construction of the accelerator and detector
2. track down problems faster when they develop
3. facilitate the construction process.

The focus of the project database, therefore, is on technical data which had to be developed in-house (see Fig. 1).

The High Energy Physics community uses highly heterogeneous computer systems. There are more than 700 PEP-II and *BABAR* collaborators at more than 70 sites worldwide. They use a mixture of Macintosh OS, MS Windows, UNIX, VMS and VM operating systems. This open environment led us to rely on the WWW for broad user access to the data.

GUI interfaces are a necessity for our users. Most of them are physicists and engineers who are only casual users of the database. They are quite capable of building their own desktop computer applications. Only GUI tools, such as WWW, can entice them from their existing disparate desktop applications onto a central database. Oracle was chosen because it is a modern, relational database; it runs on the client/server architecture our project is using, it is already site-licensed at SLAC and, in addition, is widely used at other DOE labs.

## SCOPE AND FUNCTIONALITY OF THE PROJECT-WIDE DATABASE

The core of our accelerator and detector project data is the new Oracle*CASE-generated enterprise database. This integrated database is normalized, and validation is done through constraints. Hooks are designed for other existing SLAC institutional data and for future links.

Existing legacy databases are linked to the PEP-II/*BABAR* Project Database by the Web and common data elements (examples are the Purchase Requisition database in SPIRES and flat files, the Cables database in Oracle). Through the Web pages, additional data flat files, binary and graphics files are also accessed or downloaded by our collaborators from a central place.

Presented at *The International Accelerator Database Group Workshop*,
Argonne, Ill., November 6-8, 1995

Work supported by Department of Energy contract DE-AC03-76SF00515

The following are the main subsystems, categorized by stages during construction of the machine.

*DESIGN STAGE*

1. The Personnel module.
2. The Parameter List module, which holds the physics and engineering parameters of the accelerator. Change control is applied to parameter values, and all changes are journaled in the database.
3. A Drawings module capable of producing drawing trees and bills of materials. Work is in progress to view the CAD drawings themselves on screen through WWW.
4. A Documentation tracking module

*CONSTRUCTION STAGE*

5. A Purchase Requisition Tracking system, based primarily on legacy databases.
6. A Fabrication module based on "travelers" consisting of measurements taken of the component instances. A traveler is a set of fabrication instructions and measurements to manufacture that component instance.
7. A Survey and Alignment module, which contains ideal and actual coordinates for installed components.

*INSTALLATION STAGE*

8. A Cables and Wire Lists module which is based on links to a legacy database.
9. The Inventory/Property Control system, which is based on barcodes.
10. The Environment, Safety and Health (ES&H) modules -- which includes Corrective Actions that track safety problems, and Personnel ES&H Training Records based on links to legacy database.

*PEP-II Project Database Entities & Relationships Diagram*

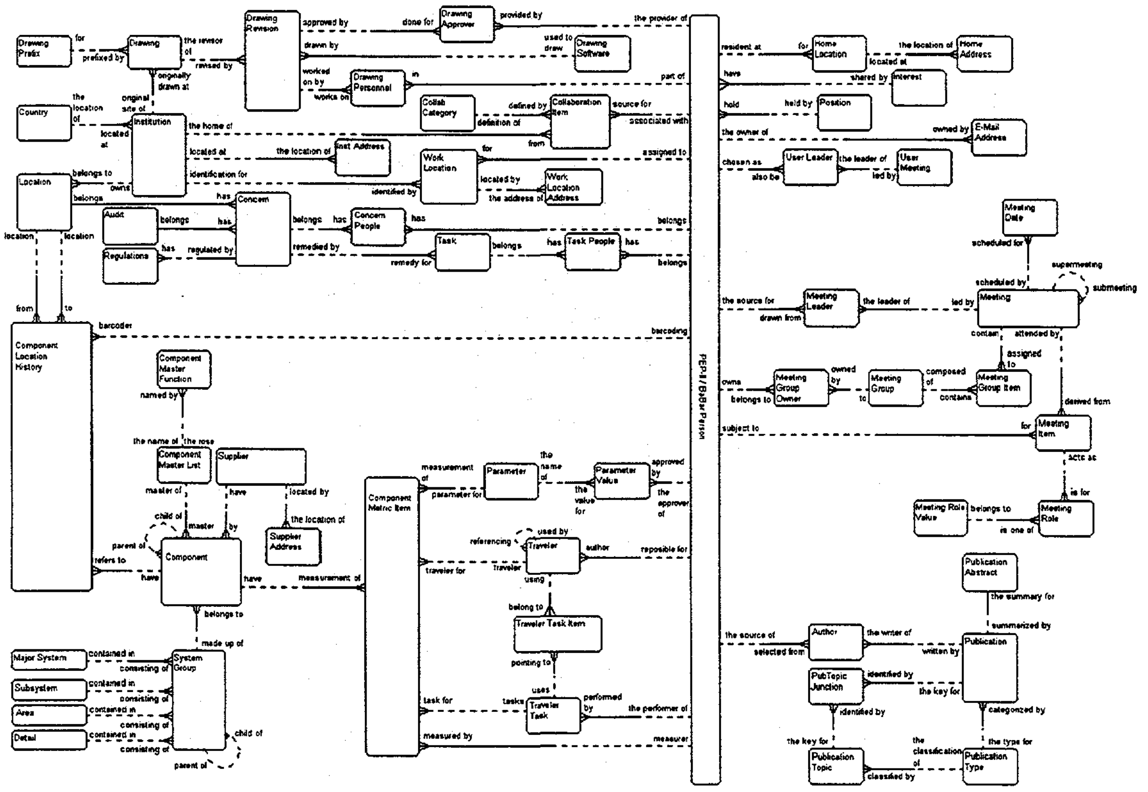


Fig. 1. PEP-II/BaBar Project-Wide Database Entities and Relationships Diagram generated by Oracle*CASE.

## *IMPLEMENTING THE KEY PIECES*

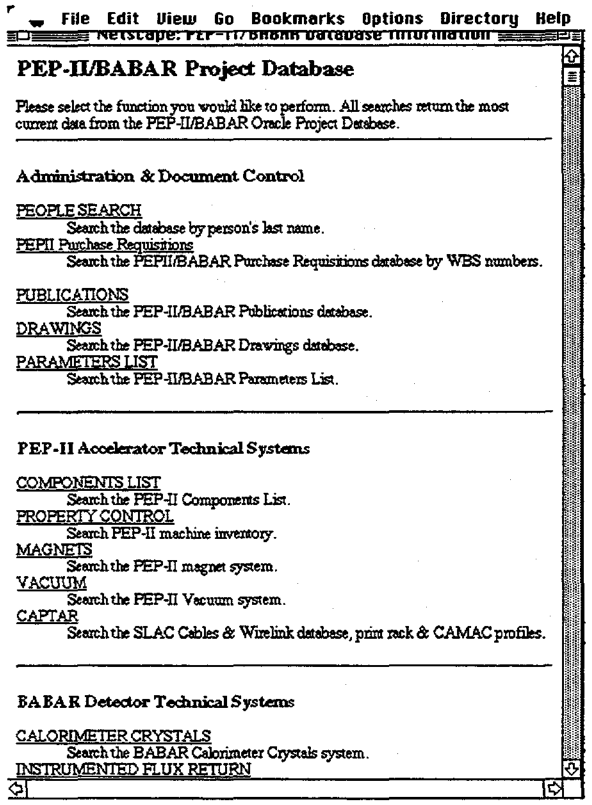
To deal with the contradiction between the large scope of an enterprise-wide database and the pressing needs of a construction project already under way, we focused on getting key pieces of the skeleton database running right away. Other pieces were created as management and production needs arose.

The first three modules implemented were (see Fig. 2):

1. Personnel (most tables in the database have relationships with this module); this includes a platform-independent e-mail distribution system.
2. Drawings and specifications (mainly, but not exclusively, CAD)
3. Components (magnets, calorimeter crystals, etc.)

The Components system was the heart of the technical part of the database, and we revised its design many times. In Fig. 3, the Entities and Relationships Diagram shows that, for a component to exist in the database, it must be entered in the Component Master List entity. The component can have many parameters and their corresponding design values. It can also have a drawing number and a revision number. Each instance of the component is entered into the Component entity. The component instance is produced according to the order of procedures from a traveler which has a traveler number and revision number. In the traveler, the fabrication instructions and measurements are identified by tasks. Measurement values are stored in the Component Metric entity. The Component Location entity records the history of physical locations of the component instance, with the final designation being its destination in the PEP-II tunnel.

We have tested this Components design with PEP-II magnet and vacuum systems, with the *BABAR* calorimeter crystals system and Instrumented Flux Return system. Through these tables, we are able to retrieve fabrication and measurement data by many criteria. Users can access data through the Web interface. Data can also be dumped from the database by third party software, like Clear Access(TM), into many other software packages familiar to, and preferred by, the users, such as Microsoft Excel (see Fig. 4 and 5).


File Edit View Go Bookmarks Options Directory Help
Netscape: PEP-II/BABAR Database Information

**PEP-II/BABAR Project Database**

Please select the function you would like to perform. All searches return the most current data from the PEP-II/BABAR Oracle Project Database.

**Administration & Document Control**

PEOPLE SEARCH
Search the database by person's last name.
PEPII Purchase Requisitions
Search the PEPII/BABAR Purchase Requisitions database by WBS numbers.

PUBLICATIONS
Search the PEP-II/BABAR Publications database.
DRAWINGS
Search the PEP-II/BABAR Drawings database.
PARAMETERS LIST
Search the PEP-II/BABAR Parameters List.

**PEP-II Accelerator Technical Systems**

COMPONENTS LIST
Search the PEP-II Components List.
PROPERTY CONTROL
Search PEP-II machine inventory.
MAGNETS
Search the PEP-II magnet system.
VACUUM
Search the PEP-II Vacuum system.
CAPTAR
Search the SLAC Cables & Wirelink database, print rack & CAMAC profiles.

**BABAR Detector Technical Systems**

CALORIMETER CRYSTALS
Search the BABAR Calorimeter Crystals system.
INSTRUMENTED FLUX RETURN

Fig. 2. Main Menu from World Wide Web

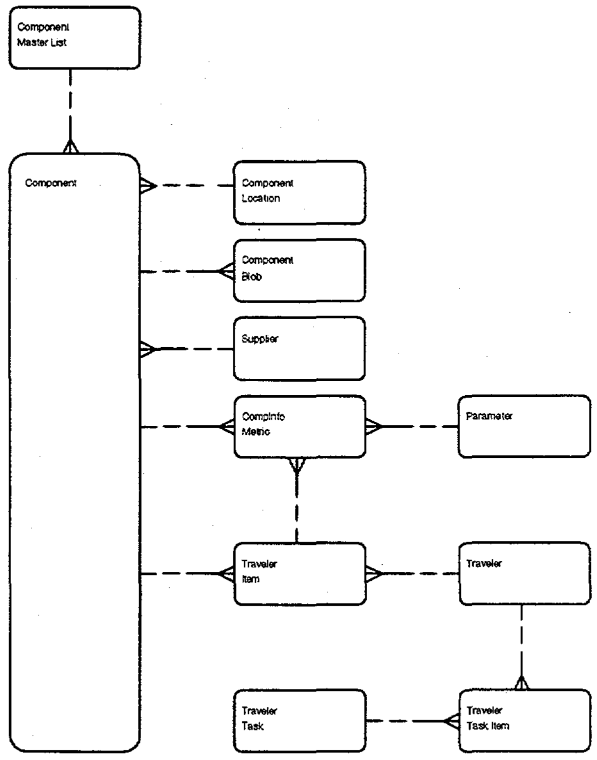
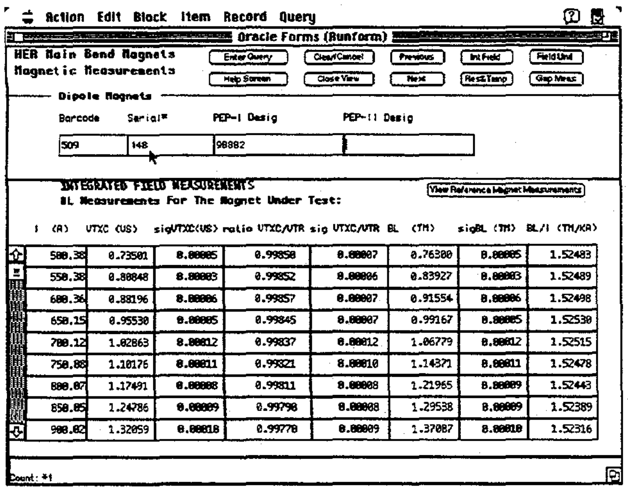


Fig. 3. Components, Travelers, Tasks and Metrics Entities and Relationships Diagram



HER Main Bend Magnets
Magnetic Measurements

Dipole Magnets

| Barcode | Serial# | PEP-I Desig | PEP-II Desig |
|---|---|---|---|
| 509 | 148 | 98882 | |

INTEGRATED FIELD MEASUREMENTS
BL Measurements For The Magnet Under Test:

| I (A) | UTXC (US) | sigUTXC(US) | ratio UTXC/UTR | sig UTXC/UTR | BL (TM) | sigBL (TM) | BL/I (TM/KA) |
|---|---|---|---|---|---|---|---|
| 500.38 | 0.73501 | 0.00005 | 0.99850 | 0.00007 | 0.76300 | 0.00005 | 1.52483 |
| 550.38 | 0.80848 | 0.00003 | 0.99852 | 0.00006 | 0.83927 | 0.00003 | 1.52489 |
| 600.36 | 0.88196 | 0.00006 | 0.99857 | 0.00007 | 0.91554 | 0.00006 | 1.52498 |
| 650.15 | 0.95530 | 0.00005 | 0.99845 | 0.00007 | 0.99167 | 0.00005 | 1.52530 |
| 700.12 | 1.02863 | 0.00012 | 0.99837 | 0.00012 | 1.06779 | 0.00012 | 1.52515 |
| 750.08 | 1.10176 | 0.00011 | 0.99821 | 0.00010 | 1.14371 | 0.00011 | 1.52478 |
| 800.07 | 1.17491 | 0.00008 | 0.99811 | 0.00008 | 1.21965 | 0.00009 | 1.52443 |
| 850.05 | 1.24786 | 0.00009 | 0.99798 | 0.00008 | 1.29538 | 0.00009 | 1.52389 |
| 900.02 | 1.32059 | 0.00010 | 0.99770 | 0.00009 | 1.37087 | 0.00010 | 1.52316 |

Count: *1

Fig. 4. Bdl measurements for HER Dipole Magnet serial number 148 from Oracle Forms4

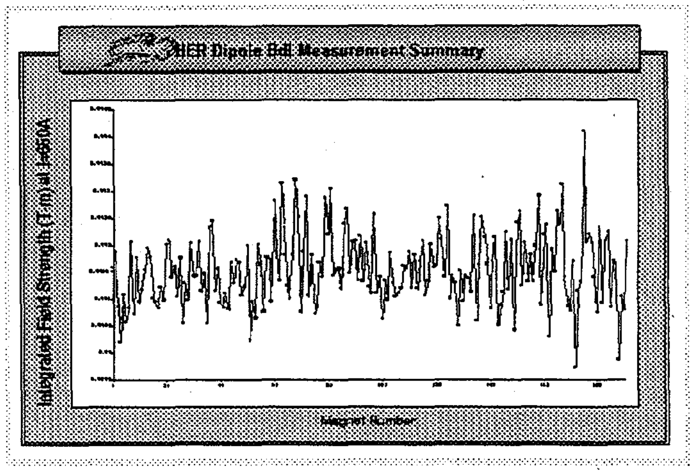


Fig. 5. Graph of Bdl measurements at 650 A for all HER Dipole Magnets from data retrieved by Clear Access and charted in Microsoft Excel.

## SOME PROBLEMS & SOLUTIONS

### *CREATING THE SEAMLESS WHOLE WITH WORLD WIDE WEB*

As many institutions have found, access to and use of the WWW has been an enormously successful approach. Providing WWW access to the many components of the PEP-II/*BABAR* Project Database has had the effect of "turning on the light" for literally hundreds of our users. Through WWW, we can provide easy searching, retrieval and reporting directly from the Oracle database. Complex search criteria, table joins and linking to legacy databases are all transparent to the users. Response time for Oracle access via WWW has been excellent.

Our goal is to provide access to all public components of the Project Database via WWW. In general, direct access to the database via Forms should only be necessary for:

1. users needing to make modifications to the database
2. tables where security issues cannot be fully resolved on WWW.

We prefer the WWW interface because we find Oracle Forms4 and Reports2 cumbersome in requiring that users run two distinct programs, even when Reports2 is called by Forms4. The two programs require a lot of computer disk space and memory, and it is a chore updating the client computers when new versions of the programs are released. Although Forms4 is nearly source code portable across platforms, we had to do significant work to manipulate buttons and fonts (see Table 1).

### *IMPLEMENTATION*

At SLAC we are currently running the CERN WWW server code on UNIX and VM computers. The VM mainframe contains our legacy databases. The UNIX WWW server is on a SUN with SQL*Net access to the Oracle database instance, which resides on an RS/6000.

The heart of the database/web interface is in one Compatible Gateway Interface (CGI) script. A major drawback to using CGI scripts is that, at this level, they do not have any security and could potentially execute undesirable commands or have unexpected results.

We are fortunate that much of our data is not of a sensitive nature. With hundreds of collaborators around the world, being able to provide information with few security constraints is a big plus. However, some data are of a semi-sensitive nature so we have been faced with the security issue. Therefore, access to some information from off-site machines is controlled.

Table 1 Comparison of World Wide Web and Oracle Forms4 GUI Interfaces

| Factors Being Compared | World Wide Web | Oracle Forms4 & Reports2 |
|---|---|---|
| Data input integrity | Data type & range checking, inter-table integrity, key-column protections & locking schemes are not built in. | **Built into Forms4.** |
| Security | Passwords have many risks still needing solutions. Several levels of security possible (server, directory, file). | **Login password provides security.** |
| ** Least requirements on computer hardware, installation, maintenance, upgrades. | **Mac & PC Web Browsers, like Netscape, need 1 Mb Hard Disk & 3 Mb RAM to run basic browser.**<br><br>**Software is free or low cost.**<br><br>**UNIX servers have Web Browsers that can run on X Window.**<br><br>**Installation or upgrade of Netscape for Mac & PC is 10 minutes. No real maintenance needed.** | Mac & PC Forms4 and Reports2 need 20 Mb Hard Disk & 16 Mb RAM. Suitable only for high end desktop computers (486, Quadra,..).<br>Programs cannot be run off the server for Mac.<br>Mac & PC can run X Window session for Forms4 & Reports2 on UNIX–but response time is not good enough.<br>If all goes well, Mac installation or upgrade of Forms4 & Reports2 takes 20 to 30 minutes. Maintenance problem if user renames hard drive, moves Oracle directory or home, etc. |
| ** Platform independent | **All files are on servers. There are consistent interfaces for queries, reports & printing.** | Fonts & icon appearances (sizes, colors) all have to be tweaked in porting cross-platform. |
| ** Spans different databases (Oracle, SPIRES, Sybase) & different file formats (flat files, help files, graphic files) | **CGI scripts can query any type of server-based database, providing links with legacy data, heterogeneous databases & different file formats.** | Much more limited. |
| Integrated query & report | **Everything is done in CGI scripts.** | Forms4 & Reports2 are two separate programs. |
| Links to third party software | **Cut/paste from Web reports page into MS Word or Excel.** | Harder to save output to any common formats for MS Word or Excel. |
| Ease of use for casual users | **No need to remember enter vs. execute query, next block vs. next record, input vs. query mode.**<br>**No need to remember/forget passwords.** | The opposite is true. |
| For our environment: | We rely on the Web for broad data query access.<br>We rely on Forms4 for the authorized people doing data input.<br><br>With only Forms4 access to the database, there were ~30 users.<br>With Web access to the database, there are now ~700 users.<br>(The Web increased our job security!!) | |

** Denotes a key factor for most users **Bold** denotes the better GUI tool.

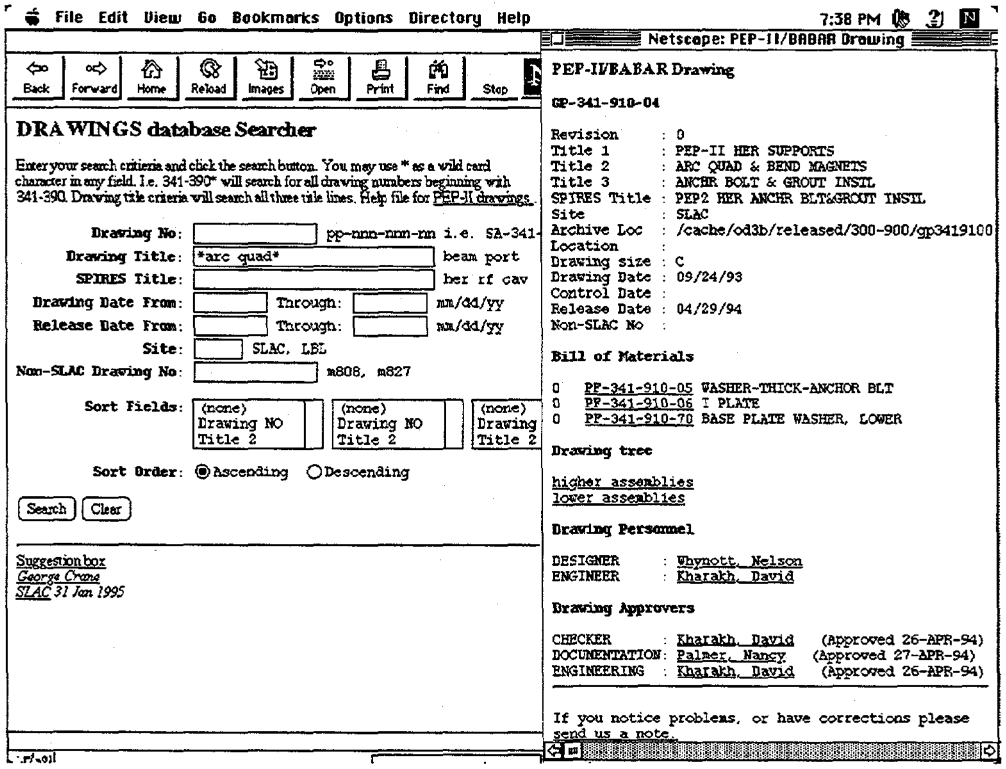
In order to provide some minimal level of security for CGI scripts on the SLAC WWW server, a CGI security Wrapper is used. The server invokes the PEP-II CGI script through the Wrapper, which is itself a CGI script. The Wrapper provides some simple checking on input to the PEP-II CGI script. This also makes it trivial to permit the execution of "authorized" UNIX commands.

The PEP-II CGI script may be called in several ways:

1. Directly from a WWW form requesting a database search with specified criteria.
2. From an http URL requesting that a customized WWW form be created based on a requesting user's IP address. In this case, the form produced may be dependent on the requesting IP address (on-site versus off-site).
3. From a hypertext reference (hot spot) requesting information from the Project Database or perhaps from a legacy database system.

The following example illustrates the process of searching for a drawing using WWW (see Fig. 5). The user first navigates to a WWW form that has several choices for search criteria, such as drawing number, title words, date ranges. The user fills in one or more search criterion fields (using a wild card character if desired) and presses the "search" button. Control then is passed to the PEP-II CGI script (via the Wrapper). Arguments passed to the CGI script include all of the filled-in search criteria plus a function parameter that tells the script which operation to perform. The arguments are examined by the script resulting in a single SQL WHERE clause. The predicate is then passed to a Pro*C program that formats and outputs data based on a declared cursor. In addition to outputting the requested data, the C program may insert http references that in turn call upon the same CGI script with other functions. For instance, engineers and approvers of the drawings are linked to our personnel tables so hot spots are included for quick reference. In addition, hot spots to produce higher and lower assembly drawing trees may be automatically created. This is one example of the many relationships that are included in the WWW pages, not only from within the Project Database in Oracle but also from hot spots referencing legacy databases (see Figs. 6 and 7).

The URL for the PEP-II/*BABAR* Project is:
**http://www.slac.stanford.edu/accel/pepii/home.html**



**DRAWINGS database Searcher**

Enter your search critieria and click the search button. You may use * as a wild card character in any field. I.e. 341-390* will search for all drawing numbers beginning with 341-390. Drawing title criteria will search all three title lines. Help file for PEP-II drawings.

```
           Drawing No: [            ] pp-nnn-nnn-nn i.e. SA-341-
        Drawing Title: [*arc quad*                 ] beam port
         SPIRES Title: [                           ] her rf cav
    Drawing Date From: [      ] Through: [      ] mm/dd/yy
    Release Date From: [      ] Through: [      ] mm/dd/yy
                 Site: [    ] SLAC, LBL
 Non-SLAC Drawing No: [            ] m808, m827

          Sort Fields: (none)        (none)        (none)
                       Drawing NO    Drawing NO    Drawing
                       Title 2       Title 2       Title 2

           Sort Order: (●) Ascending  ( ) Descending
```

Suggestion box
*George Crane*
*SLAC 31 Jan 1995*

**PEP-II/BABAR Drawing**

```
GP-341-910-04

Revision      : 0
Title 1       : PEP-II HER SUPPORTS
Title 2       : ARC QUAD & BEND MAGNETS
Title 3       : ANCHR BOLT & GROUT INSTL
SPIRES Title  : PEP2 HER ANCHR BLT&GROUT INSTL
Site          : SLAC
Archive Loc   : /cache/od3b/released/300-900/gp3419100
Location      :
Drawing size  : C
Drawing Date  : 09/24/93
Control Date  :
Release Date  : 04/29/94
Non-SLAC No   :

Bill of Materials

0   PF-341-910-05 WASHER-THICK-ANCHOR BLT
0   PF-341-910-06 I PLATE
0   PF-341-910-70 BASE PLATE WASHER, LOWER

Drawing tree

higher assemblies
lower assemblies

Drawing Personnel

DESIGNER      : Whynott, Nelson
ENGINEER      : Kharakh, David

Drawing Approvers

CHECKER       : Kharakh, David   (Approved 26-APR-94)
DOCUMENTATION: Palmer, Nancy    (Approved 27-APR-94)
ENGINEERING   : Kharakh, David   (Approved 26-APR-94)

If you notice problems, or have corrections please
send us a note.
```

Fig. 5 Drawings Web HTML Form, and Report which includes more "hot spots"

*OTHER USER INTERFACE ISSUES*

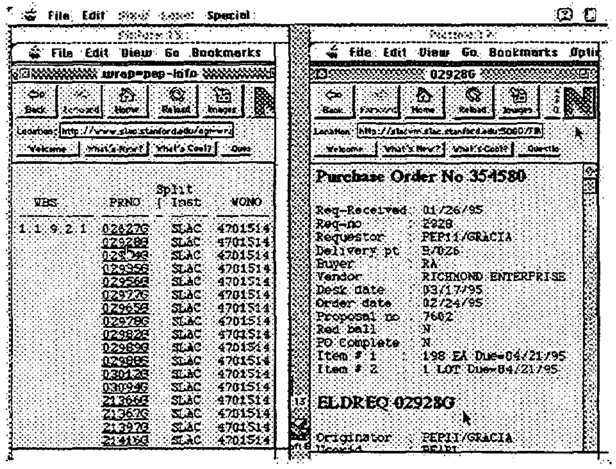
One of our most successful applications is the Purchase Requisition tracking module which integrated together Oracle tables and six legacy databases (see Fig. 6). Before this, users had to search for the data using these disparate databases. Through WWW, we provided the interface into these hybrid systems as part of our project enterprise database. In the beginning, we were reluctant to venture into this legacy swamp (saying, as most sensible programmers do, that such a messy system should be re-engineered first). However, this turned out to be one of the most utilized modules and encouraged us to look at the enterprise-wide database more from the user viewpoint. That is, while the database means solid design of entity relationships and table definitions to the programmers, to the users the screens and reports interface *is* the database. The users are not concerned with how many Oracle and SPIRES databases (often in messy legacy systems) a Web interface has to span, nor need they be.

Our users capture data in a myriad of ways, initially not involving Oracle–(in spreadsheets, flat files, non-relational databases)–and they will continue to do so. We work with the users to record the data in these formats (e.g., in MS Excel spreadsheet templates), and write UNIREXX, C or SQL*Loader programs to load this data into Oracle. Such measures for user data input that help free the users from the underlying database table structures are popular within our project.

*FAST TRACK*

We have a very small team doing the analysis and development (averaging two and a half full-time people over 3 years). The physicists and engineers were extremely busy with design problems of their own to meet the requirements of the project. Time and again, the users told us that they did not have a clear idea of what they wanted. In this hectic environment, it was often difficult to perform a detailed analysis of the system. We combined lessons we learned during our 1992 visit to CERN in Geneva, Switzerland, our knowledge of databases used to track the manufacture of other products, and what our users liked and disliked. We have learned to quickly give key pieces of the database to the users so that they can test the interface and the data. The rapid turn-around has been important for maintaining user involvement and management support.

This prototyping worked extremely well. Users were eager to state how the system should be improved to better meet their needs. CASE has been an important tool in shortening this development cycle.


File Edit Special
File Edit View Go Bookmarks
wrap=pep-info
Back Forward Home Reload Images
Location: http://www.slac.stanford.edu/cgi-wrap
Welcome What's New? What's Cool? Ques
File Edit View Go Bookmarks Optio
029286
Location: http://slacvm.slac.stanford.edu:5080/FI

```
                Split
  WBS       PRNO  [ Inst    WONO
1.1.9.2.1  026276   SLAC  4701514
           029286   SLAC  4701514
           029346   SLAC  4701514
           029356   SLAC  4701514
           029566   SLAC  4701514
           029776   SLAC  4701514
           029656   SLAC  4701514
           029786   SLAC  4701514
           029826   SLAC  4701514
           029896   SLAC  4701514
           029886   SLAC  4701514
           030126   SLAC  4701514
           030946   SLAC  4701514
           213666   SLAC  4701514
           213676   SLAC  4701514
           213976   SLAC  4701514
           214156   SLAC  4701514
```

**Purchase Order No 354580**

```
Req-Received  : 01/26/95
Req-no        : 2928
Requestor     : PEP11/GRACIA
Delivery pt   : B/026
Buyer         : RA
Vendor        : RICHMOND ENTERPRISE
Desk date     : 03/17/95
Order date    : 02/24/95
Proposal no   : 7602
Red ball      : N
PO complete   : N
Item # 1      : 198 EA Due=04/21/95
Item # 2      : 1 LOT Due=04/21/95
```

**ELDREQ 029286**

```
Originator    : PEPII/GRACIA
```

Fig. 6. World Wide Web interface to PEP-II Purchase Requisitions joining Oracle and legacy SPIRES data

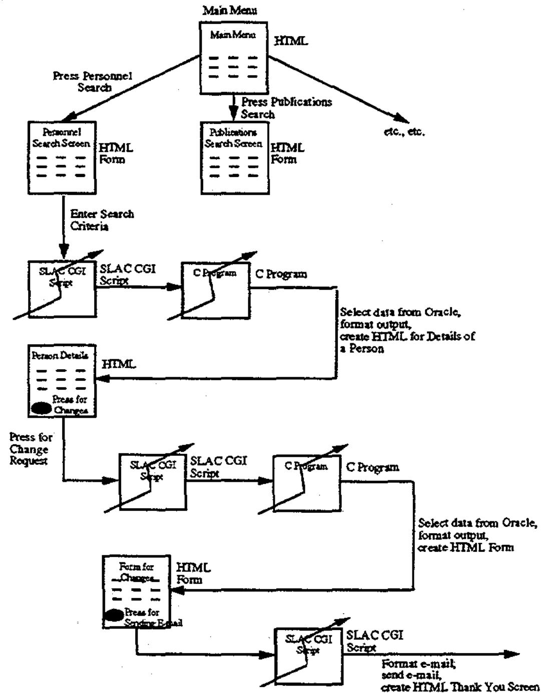


Fig. 7. Flow Chart of Web Database Queries to Send E-mail for Updating Personnel Information

*USING CASE FOR DESIGN AND MAINTENANCE*

The project is a large one with many database objects as well as many screens and reports. We needed something to help us manage these and, just as importantly, we wanted to gain experience with Oracle*CASE. CASE has helped greatly, but at times has hindered us from upgrading client software. CASE generators for Forms4 were not released at the same time as the CDE suite. We had no choice but to abandon CASE for a time.

The first lesson one learns about CASE is that training in the use of the product is a prerequisite. This training is crucial to understanding how facets of CASE integrate and complement each other. The course materials one obtains from completing CASE classes include recipes detailing how to accomplish the required operations to develop a project. Unfortunately, the courses does not include how to recover from missteps, and on the first project these will be made. After a short while, the workings of CASE become familiar and its benefits apparent.

CASE allowed us to build the original screens and reports rapidly, and to easily implement changes. Using CASE, we could often produce in a couple of hours what might have taken a week or more using only Forms and Reports. CASE did not solve all our forms problems. We had to code some procedures on our own, as well as some triggers. However, approximately 90% of the code was written by CASE. The code produced also served as a PL/SQL training aid. By examining the code that CASE produced, developers were able to write better triggers in other applications.

In the prototyping methodology where objects and the relationships between them will change, CASE provided some stability and documentation. This project was first built against a version 6 database when roles were unavailable; however the fact that all the grants were stored in CASE made it simple to restore permissions on dropped and recreated objects.

## FUTURE PLANS

Work will be done to expand to more accelerator and detector components.

This construction and history data will be useful for the ongoing maintenance of the *B* Factory facility. We plan to make links to legacy SLAC databases in order to facilitate this function.

## RE-USABILITY

Many parts of the CASE-generated database design can be re-used. In particular, the component and traveler system is generic and we are able to apply it for both the accelerator and the detector. Being able to re-use different parts of the database design with Oracle*CASE has in fact allowed us to cover a broad scope with a small team.

## SUMMARY

A Graphical User Interface that is intuitive and easy to use is a key to the PEP-II/*BaBar* Project Database being widely adopted by the Project Management and the general user community. WWW access to the data has greatly helped (linking even different database software).

Delivery time of the software had to be fast to maintain user involvement. Oracle*CASE has been an important tool for this rapid prototyping, enabling our small team to build an enterprise database.

This project-wide system is a useful tool to maintain quality assurance during the construction of PEP-II and *BaBar*. It is also helping to facilitate management and coordination of the collaborating labs. This integration of administrative and technical data is an innovative use within the accelerator community of a central project-wide database.

The vision and support of the PEP-II/*BaBar* Project Management in recognizing the need for a centralized data repository is a major factor in bringing this about.

## REFERENCES

[1] A. Chan, S. Calish, G. Crane, I. MacGregor, S. Meyer, A. Weinstein, J. Wong, 'The PEP-II Project-Wide Database', Proceedings of the 16th IEEE Particle Accelerator Conference (PAC95)

[2] J. Poole, 'Databases for Accelerator Control - An Operations Viewpoint', Proceedings of the 16th IEEE Particle Accelerator Conference (PAC95)

## ACKNOWLEDGMENTS

We are grateful for the support and vision that Jonathan Dorfan, Geoff Girvin, David Hitlin and Michael Zisman gave us in 1992 which launched this exciting enterprise-wide effort. We also thank the present PEP-II/*BABAR* Project Management for their continued support, the Data Administrators of the different systems for helping with the design and for maintaining the integrity of their thousands of records, and our colleagues at CERN and SLAC from whom we learned much.

*This work was supported by the U.S. Department of Energy, under contract numbers DE-AC03-76SF00515.