

SS-2001-03: The TRACS Web Database

Daniel Mossop

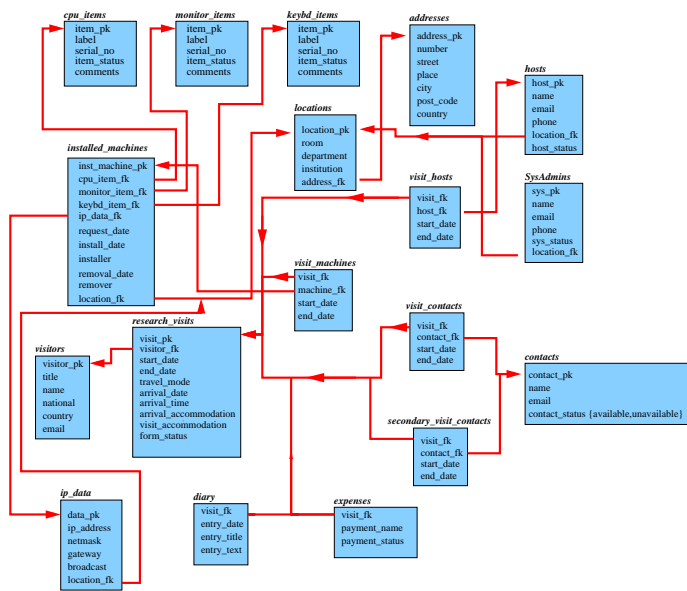
EPCC, University of Edinburgh
University of Edinburgh
email dagm@dcs.ed.ac.uk

Introduction

EPCC has been coordinating the TRACS (Training and Research on Advanced Computing Systems) programme since 1993 [1]. TRACS allows EC-resident academic and industrial researchers to come to EPCC for visits lasting between four to thirteen weeks. During this time they can use EPCC resources and consultancy to help their research. As part of the programme visitors are given access to EPCC-owned terminals, which are distributed to host departments, mainly located in Edinburgh, for the duration of their visit. The Web Interface was started to address the need for a method of maintaining up-to-date records detailing the whereabouts of these terminals. It has since been expanded to hold a wider range of data relating to the TRACS visits.

Database Structure

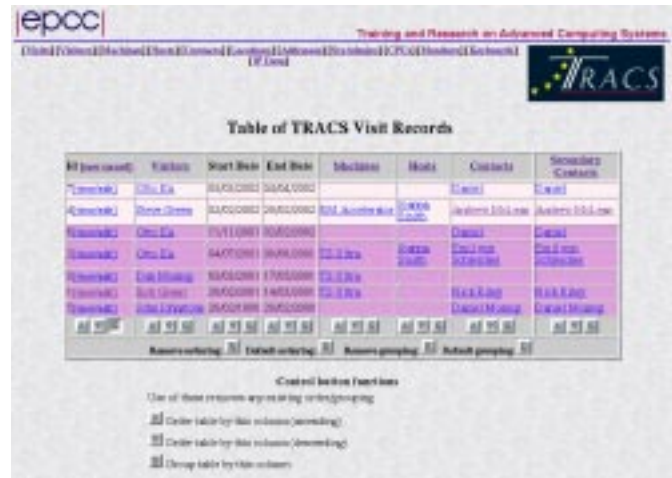
The back-end of the TRACS Web Database system is a MySQL database. The database allows a wide range of TRACS related data to be stored. To meet the original requirement of the system EPCC terminals, their component parts and details of their installations can all be stored in the database. The database also contains tables to hold details of the TRACS visitors and their visits. The database allows the dates of visits and installations to be recorded, which enables the database to store not only their current status, but also their history. The current form of the database is shown in the diagram below:



The TRACS database.

General Table Generation

One requirement of the TRACS Web Database is that its front-end should provide a means of viewing a summary of subsets of the TRACS data. This has been implemented with Perl scripts that use the CGI to output HTML code that can be viewed on web browsers. The method chosen to display the data is to format them in dynamically generated tables. In order to minimise the amount of code required to implement this, all the necessary tables are generated by a general routine. As well as displaying the data, this routine also provides features to help improve the clarity of the data. These include buttons to control the ordering of the columns, and shading to show the past, present and future records. Access to the database is made using the Perl Database Interface (DBI) [2]. The DBI allows the development of database accessing routines that are independent of the actual database implementation. This enables the front-end implementation of the web interface to be independent of the back-end implementation, increasing the code portability.



An example table created by the general table routine.

General Record Manipulation

A second requirement of the TRACS Web Database is that the front-end should allow records to be handled on an individual basis. As well as enabling the user to view individual records, it should also make it possible for the administrator to edit the records and create new ones. Again to keep to a minimum the amount of code required, this has been implemented using a general routine.



An example table created by the general table routine.

Different Views

Due to the wide scope of the data that can be stored in the database, the web database actually has several different classes of user. An example of such a class is the systems user who is concerned with maintaining details of the EPCC terminals, but not with the TRACS visitors. The Web Database accommodates this by accepting different logins, using htaccess, for each class and only allowing access to the data relevant to that user class.

Acknowledgements

Thanks to Elson Mourao who was responsible for the original database design. Thanks also to Mario Antonioletti and Elena Breitmoser who developed the initial implementation of the Web Database and who supervised this project.

References

1. The TRACS Home Page - <http://www.epcc.ed.ac.uk/tracs>.
2. Perl in a Nutshell (Chapter 12) - Siever, Spainhour and Patwardhan. O'Reilly 1999