Proceedings World Geothermal Congress 2000
Kyushu - Tohoku, Japan, May 28 - June 10, 2000

DATABASE DEVELOPMENT FOR INTEGRATED RESERVOIR MODELING

J. L. Stevens[1], J. W. Pritchett[1], S. K. Garg[1], S. Nakanishi[2], K. Ariki[3], and S. Yamazawa[4]
[1]Maxwell Technologies, Systems Division, 8888 Balboa Ave., San Diego, CA 92123-1506, USA
[2]Electric Power Development Co., Ltd., 6-15-1 Ginza, Tokyo 104-8165, Japan
[3]Mitsubishi Materials Corporation, 1-297 Kitabukuro-cho, Omiya, Saitama 330-8508, Japan
[4]New Energy and Industrial Technology Development Organization, 3-1-1 Higashi Ikebukuro, Tokyo 170-6028, Japan

Key Words: geothermal, database, reservoir modeling

ABSTRACT

A database system is being developed that both stores archival geothermal data and supports geothermal reservoir modeling calculations. The database maintains field data and helps to organize and review input and output data from calculations. The initial design is based on GEOSYS, a geothermal database system originally developed for Unix workstations, which is being redesigned, extended to support calculations, and converted to Microsoft Windows (95/98/NT). Development costs are being reduced through incorporation of database technology, development tools, and third party components that are available for Windows. The use of ActiveX Data Objects (ADO) allows the system to run using a local Microsoft Access database or a client-server relational database such as Oracle or SQL server with no changes to code.

1. INTRODUCTION

The New Energy and Industrial Technology Development Organization (NEDO) initiated a program in 1997 entitled "Development of Technology for Reservoir Mass and Heat Flow Characterization." This project is divided into two major efforts: 1) "Characterization of the Hydrological Properties of Fractures", and 2) "Monitoring and Modeling of Reservoir Mass and Heat Flows." The "Monitoring and Modeling of Reservoir Mass and Heat flows" entails (a) improving available techniques for making precise measurements of changes in gravity, resistivity, self-potential, and seismic properties which occur when geothermal fluid is withdrawn from or injected into geothermal reservoirs, and (b) developing the technology for carrying out integrated analyses of these varied data sets with more conventional engineering data, incorporating numerical reservoir modeling, so that more robust reservoir models can be developed which will help optimize exploitation strategies and minimize end-user cost for geothermal electricity. NEDO has entrusted the modeling part of the project jointly to the Electric Power Development Company ("EPDC"—a public power plant construction company located in Tokyo which is the developer of Onikobe and Oguni geothermal fields) and to Mitsubishi Materials Corporation ("MMC"), owner/operator of Ohnuma and Sumikawa geothermal fields in the Sengan area of northern Honshu.

One component of the EPDC/MMC effort involves the development of computer software. EPDC and MMC have asked Maxwell Technologies, Inc., to provide services in connection with this effort. One part of this effort is development of a database system designed to maintain field data, organize calculations and support comparisons of data and calculations. A feasibility study performed at the start of this project (Stevens and Patnaik, 1998), recommended development of the database system on a Microsoft Windows platform, using as a starting point the database in GEOSYS, a geothermal database system originally designed for Unix operating systems. The first phase of this project has focused on development of low level database software which is required by most other software, and development of prototypes intended to resolve the most critical issues first, specifically issues relating to database access, graphics, and user interface. The second phase of this effort is focusing on design and development of the user interface, and development of procedures to allow extension of the software to easily incorporate new data types and for comparison with calculations.

2. DATABASE SYSTEM DESIGN

The objective of this project is to develop a database system that both stores archival geothermal data and supports calculations being performed by the Integrated Reservoir Modeling System. The database will maintain field data and help to organize and review input and output data from calculations. This paper describes the overall architecture of the system, basic functionality, database structure, and graphical displays. In addition, we discuss the design decisions that needed to be made early in the project, and less critical design options that will be decided during the course of development.

The initial design of the system is being developed using the basic functionality of GEOSYS (version 1.0 described in Stevens et al, 1992) converted to a PC running 32 bit Windows (Stevens and Patnaik, 1998), with the addition of support for most of the information and displays in the reservoir modeling studies of Pritchett et al (1998, 2000). The combination of these two pieces contains the essential elements needed for the database system.

Geothermal data types supported by GEOSYS include:

1. Surface geographic information including Maps and images, well locations, overlays of general geographic information.
2. Subsurface geographic information including well deviations and stratigraphy, which can be displayed in cross section plots.
3. Well logs (temperature, pressure, spinner).
4. Well completion information.
5. Well test information.
6. Production time history, including chemical and physical measurements.

New data types that are being added as part of the current project include gravity, self-potential, resistivity, seismic data, and seismic properties (material velocity, density, and attenuation properties). The database is also being modified to support calculated quantities that can be compared directly with observations.

The database system is being constructed using a three-tiered design. These three tiers can be described as:

1. Data – relational database tables and files in which all of the information about the geothermal reservoir and calculations are maintained.
2. Analysis – all database queries, file reading/writing operations, and all calculations and analysis that are performed on the data.
3. User interface – creates all user displays and maintains all interactions with the user, responding to mouse clicks and keyboard.

Separating the system into these three parts makes the system easier to maintain. Note that these are logical, not physical, tiers, and they may exist all on one computer, or be distributed over more than one computer. The user interface interacts with the analysis modules, and only rarely interacts directly with the database (database input forms and ad-hoc queries being the primary exceptions). The analysis modules contain no user interface code, but have methods to provide information to the user interface. The data tier contains only structured data, and does not contain any code.

User interface design has evolved substantially over the past several years, aided by improvements in operating system support. In particular, modern user interfaces give more power to the user and interfere less with normal workflow by keeping commonly used options easily available, and by minimizing the number of dialog boxes that the user must respond to. For example, the original version of GEOSYS consisted of eight independent programs, each of which had a number of dialog boxes for selecting wells and data types. In the new database system, there is a distinct document type, each accessible from a single program, that corresponds to the different modules of the earlier version. Also, some of the information that previously required dialog boxes is now available in the form of floating toolbars (floating toolbars can be moved on the screen and either displayed or hidden by the user). Figure 1 shows a view of a geothermal well at the Sumikawa geothermal field, together with circulation loss information and a set of temperature logs. The field and well can be selected from the drop down list boxes on the toolbar, so the user can easily and quickly see a similar view of any well in the database.

3. DESIGN ISSUES

One of the more difficult aspects of modern software development is the rapid change in operating system capabilities and features. This is in most respects a good thing, because the changes, at least as a general rule, increase capabilities, and the software tools that support them simplify code development. However, it also means that software design is a moving target – software must be able to evolve and change with operating system improvements, or it quickly becomes obsolete and outmoded. Consequently, it is important to anticipate future operating system changes, to experiment with multiple ways of accomplishing tasks, and to modularize code to simplify maintenance.

There are several aspects of the system that are being tested and evaluated in order to make design decisions. These fall into the following categories:

3.1 Development Compiler and Tools

Most of the system is being developed using Microsoft Visual C++, version 6, with some modules being developed in DEC Fortran, version 6. Visual C++ provides a standard structure so that programs written with it maintain a look and feel similar to other Windows programs. It provides basic support for the document/view architecture that separates data and user interface, and provides basic support for graphics and printing. It also provides basic support for OLE, which allows programs to contain documents generated by other programs. For example, the graphics generated by the database system can be embedded in a Word document using OLE.

These compilers are excellent development tools for Windows programs and generate high performance code with full access to all Windows capabilities. However, they are also complex and sometimes difficult to program, and some parts of the program, particularly interactive database input forms, are better developed using Visual Basic or Microsoft Access. Figure 2 shows the data entry and review module which has been developed using Visual Basic. All of the data can be found easily from the data selection form. The user selects a geothermal field and well, if appropriate, and the table of interest. All of the corresponding records are displayed in a summary format. The user can then select a record for editing or review, or add new records. Unique key values for each record are generated automatically and the database maintains links between tables without requiring any action on the part of the user.

In addition, we are using Microsoft Visual Interdev and Visual J++ (Java compiler) to experiment with a web browser based interface. The latest generation of web browsers support XML (extended markup language), DHTML (dynamic HTML, also called Javascript), and Java. These tools allow development of web pages with much better interactive capabilities than earlier browsers. The advantage of a browser based system is that the database and analysis modules are stored on a centralized system in a single location, and it is not necessary to install the program on individual computers. While browser based programs do not have all of the capabilities of a Windows program, they are useful as an auxiliary interface for many common tasks. Figure 3 shows a form for quickly reviewing the information in a database table using a web browser.

3.2 Database and Database Access Method

The system is being designed initially to use a Microsoft Access database, however we are also testing the system using an Oracle database. The database programming interface uses OLEDB, a relatively recent Microsoft technology, with a higher level interface referred to as ActiveX Data Objects (ADO), now in version 2.1. ADO is a better choice than other methods (for example, the earlier DAO (Data Access Objects) and ODBC access methods) since it has higher performance drivers, and is the database access method that Microsoft will be using in the future in all of its products. In our initial experiments with ADO, we have found that we can use it successfully with all of the development tools, and that it is generally simpler than the earlier methods, particularly for accessing multiple types of databases. We have successfully retrieved data from databases in Microsoft Access and Oracle with only minor changes in code.

3.3 Modularization

Windows programs can be broken down into separately maintainable pieces in several ways. Typically, a program consists of a main program and a number of static libraries or dynamic link libraries (dll's) that are called by the main program. However, with some additional programming, dll's can be turned into COM objects, which are then more easily reusable because they can become essentially language independent modules that can be called easily by Visual Basic and other applications. The Active Template Library distributed with Visual C++ makes creation of COM objects somewhat easier, but they still add enough additional complexity to the programming task, that it is important to evaluate when and whether creation of COM objects is worthwhile.

3.4 Graphics

A variety of methods are being tested for generating graphical displays. We are using more than one method for creating the graphics modules, so that we can apply the best tool to each task. We are currently testing the following methods for generating graphics, and may expand this list if other commercially available programs are found to have significant advantages.

1. Windows graphics primitives – while tedious to program, generating graphics directly from graphics primitives allows full control over the graphics display, and in some cases is the easiest and/or most efficient way to produce custom graphics. Figure 4 shows a map of the Sumikawa geothermal field and geographic overlays generated with graphics primitives.
2. Graphpac – a relatively simple and portable system used in part of the X Window version of GEOSYS. This package has recently been converted to support Windows graphics as well as X and Postscript graphics. It has the advantages of simple conversion of existing graphics, and good low level control over the graphics displays. The display of the Well shown in Figure 1 was generated with Graphpac.
3. Olectra Plot – this is a redistributable commercial plotting package that is used as an ActiveX control (a type of COM object). It is a fairly capable two and three dimensional plotting package that supports a variety of graph types. Figure 5 shows a contour/bedsheet plot of travel times through a geothermal field generated with Olectra plot.
4. Tchart Plot – this is a plotting package similar to Olectra Plot, but with some better methods for displaying line charts, and OpenGL support for doing 3D graphics. Figure 6 shows a time history of production measurements generated with the Tchart plot program.

5. DEVELOPMENT PLANS

The first stage of database development focused primarily on development of low level database access functions, and prototyping to resolve design issues relating to database access, graphical displays, and user interface. Most database access issues have been resolved, and the next phase of the project focuses on development of the user interface, development of middle tier components for connecting the database and data to the user interface, and additional development of graphical displays. An important part of the next design phase is to identify ways to minimize the amount of code development required through a combination of the use of commercial components and development of a well organized set of middle tier components. The first phase of the project concentrated on important low level details and design decisions. A prototype containing more complete modules is now under development.

ACKNOWLEDGEMENT

NEDO supported this work as part of the "New Sunshine Project" sponsored by the Agency of Industrial Science and Technology (AIST) of the Japanese Ministry of International Trade and Industry (MITI).

REFERENCES

Pritchett, J. W., J. L. Stevens, J. Combs, and S. K. Garg (1998), "Utilization of nonconventional data sets for geothermal reservoir monitoring: a computational feasibility study," Maxwell Technologies technical report submitted to NEDO/EPDC/MMC, MSD-DTR-98-16048, March.

Pritchett, J., J. L. Stevens, P. Wannamaker, S. Nakanishi, and S. Yamazawa (2000), "Theoretical Feasibility Studies of Reservoir Monitoring Using Geophysical Survey Techniques," *Proc. World Geothermal Congress 2000*, Japan (in press).

Stevens, J. L. and P. C. Patnaik (1998), "Software development plan to transfer database, reservoir simulator, and postprocessors to a PC environment," Maxwell Technologies technical report submitted to NEDO/EPDC/MMC, MSD-DTR-98-16068, March.

Stevens, J. L., L. Luu, S. K. Garg, T. G. Barker, and J. W. Pritchett (1992), "GEOSYS: A prototype interactive data management system for geothermal analysis (phase 2)", S-CUBED final report submitted to Kyokuto Boeki Kaisha, Ltd., for New Energy and Industrial Technology Development Organization, SSS-FR-92-13423, March.

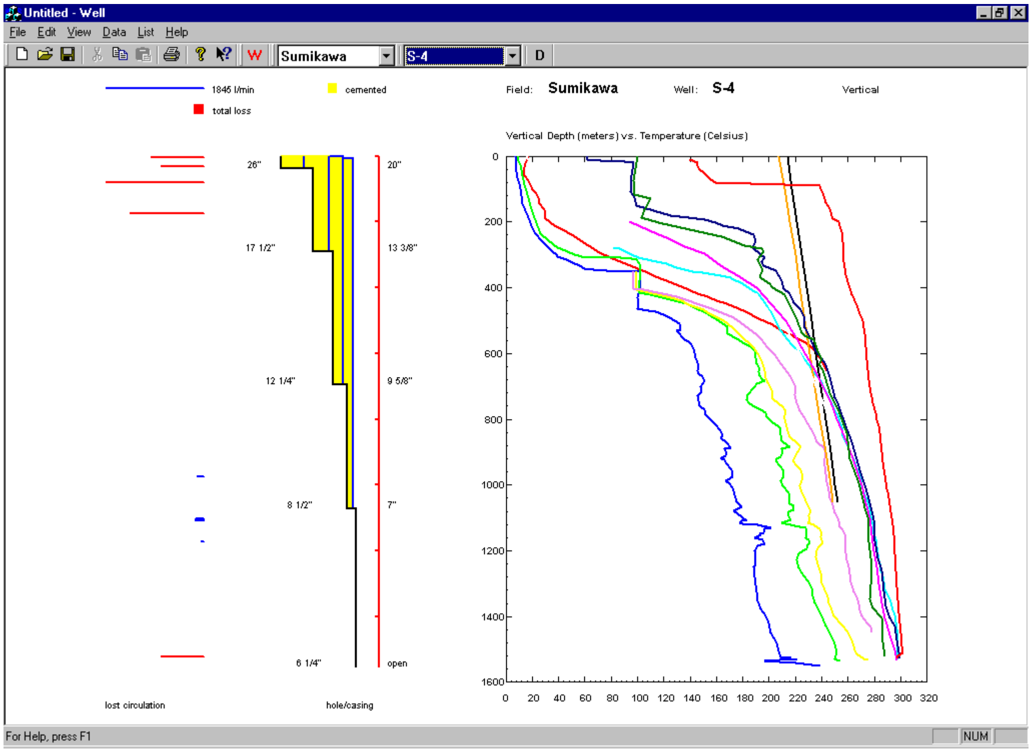


Figure 1. Module showing detailed structure of a geothermal well, circulation loss, and temperature logs. The field and well can be selected from the dropdown list on the toolbar, so the user can quickly see a similar display for any well. Several types of well log data can be displayed.

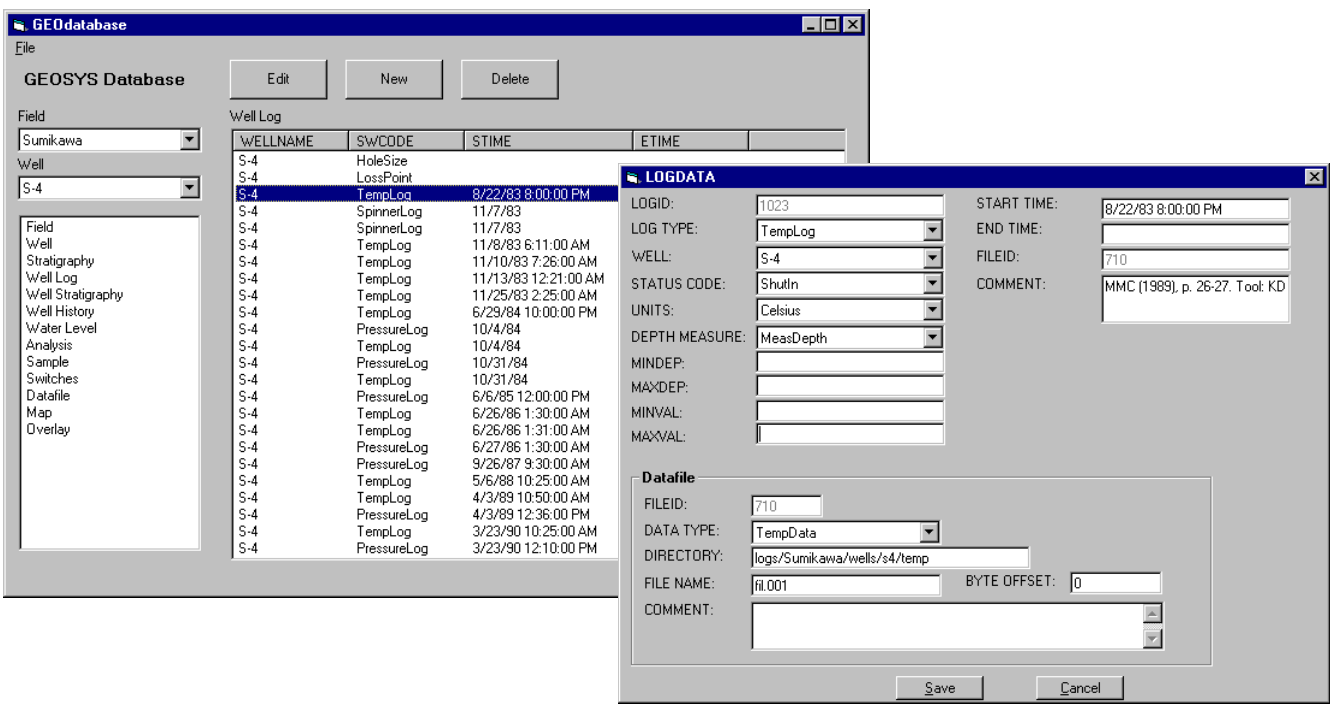
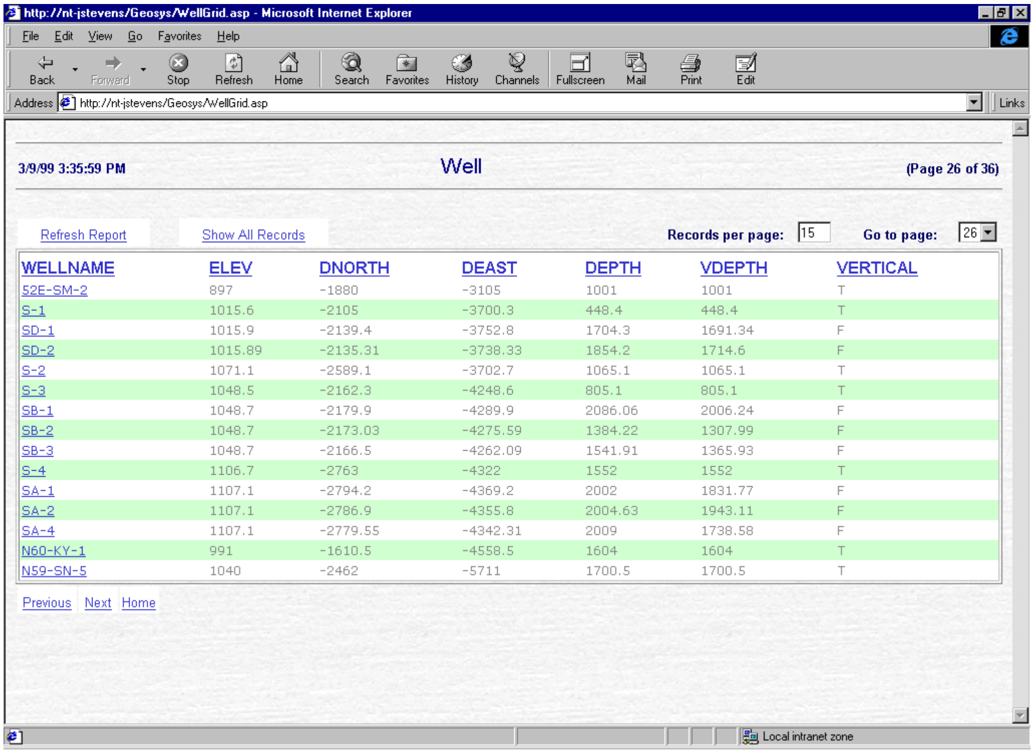


Figure 2. Forms for data entry and review. Data in all database tables can be selected from the form on the left. A detailed form is presented for editing, data entry, and review. Key values are generated automatically and are visible in the form but cannot be changed by the user. Units and data types are selected from a drop down list.



WELLNAME	ELEV	DNORTH	DEAST	DEPTH	VDEPTH	VERTICAL
52E-SM-2	897	-1880	-3105	1001	1001	T
S-1	1015.6	-2105	-3700.3	448.4	448.4	T
SD-1	1015.9	-2139.4	-3752.8	1704.3	1691.34	F
SD-2	1015.89	-2135.31	-3738.33	1854.2	1714.6	F
S-2	1071.1	-2589.1	-3702.7	1065.1	1065.1	T
S-3	1048.5	-2162.3	-4248.6	805.1	805.1	T
SB-1	1048.7	-2179.9	-4289.9	2086.06	2006.24	F
SB-2	1048.7	-2173.03	-4275.59	1384.22	1307.99	F
SB-3	1048.7	-2166.5	-4262.09	1541.91	1365.93	F
S-4	1106.7	-2763	-4322	1552	1552	T
SA-1	1107.1	-2794.2	-4369.2	2002	1831.77	F
SA-2	1107.1	-2786.9	-4355.8	2004.63	1943.11	F
SA-4	1107.1	-2779.55	-4342.31	2009	1738.58	F
N60-KY-1	991	-1610.5	-4558.5	1604	1604	T
N59-SN-5	1040	-2462	-5711	1700.5	1700.5	T

Figure 3. Well database table shown in a Web browser in a gridded format. The grid can be ordered by any column. Clicking on the well name brings up a more detailed form for the well. With the database placed on a Web server a single copy of the database can be centrally maintained and accessed by any authorized user.

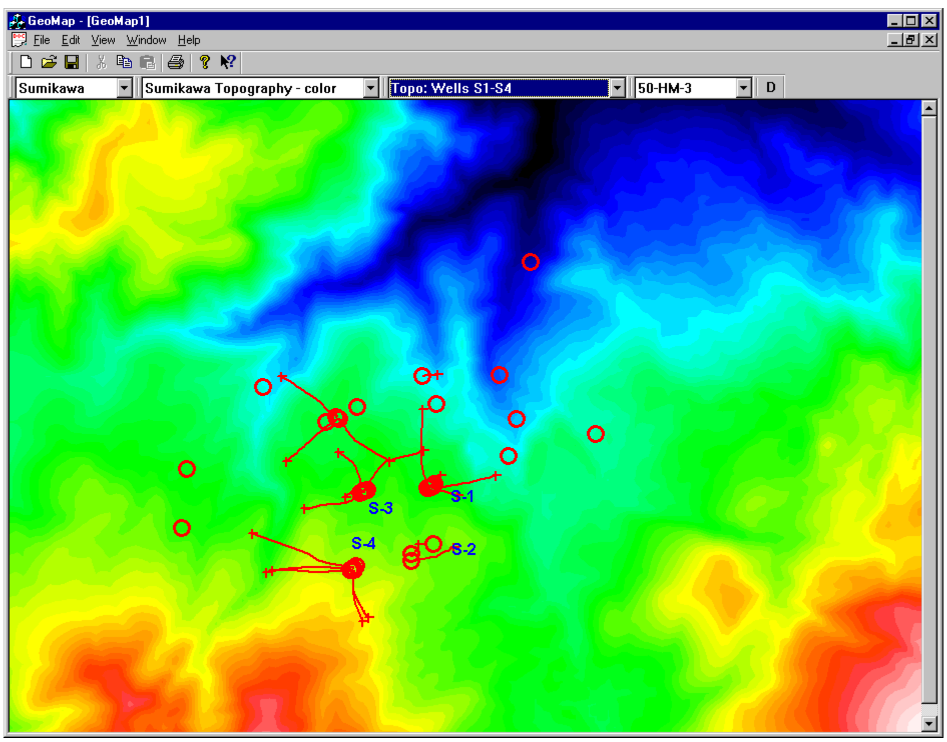


Figure 4. Color/grayscale contour plot of the Sumikawa geothermal field elevation displayed in the prototype Map module, with overlays well locations, well deviations, and the names of major wells.

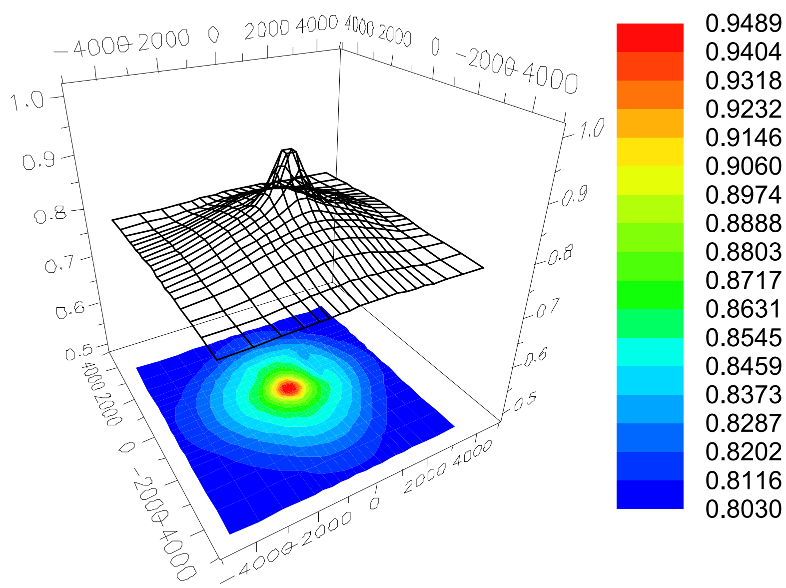


Figure 5. Contour/bedsheet plot made using Olectra chart showing seismic travel times through a STAR calculation with a thickness of 3000 meters.

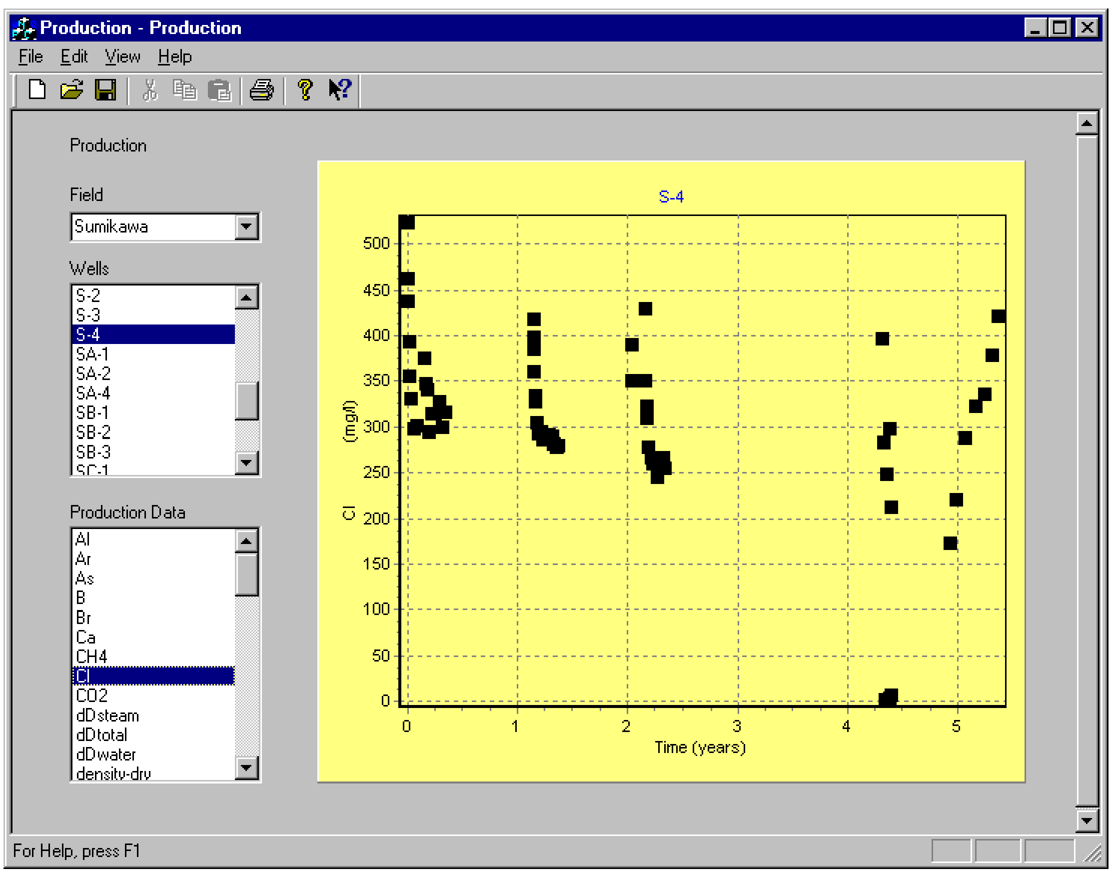


Figure 6. Prototype production module showing Chloride measurements in a geothermal well – here the user selects the field, well, and data type on the left, and a time series of measurements is displayed on the right.