2006-1923: A HOMEWORK PROBLEMS DATABASE: DESIGN AND IMPLEMENTATION

Matthew Roberts, University of Wisconsin-Platteville

MATTHEW ROBERTS is an Assistant Professor in the Department of Civil and Environmental Engineering. Dr. Roberts earned his B.S. in Civil Engineering from Brigham Young University in 1993 then spent four years in the U.S. Air Force as a civil engineering officer. He received his Ph.D. from Texas A&M University in 2002 and has been teaching structural engineering topics at the University of Wisconsin–Platteville since then.

Christina Curras, University of Wisconsin-Platteville

CHRISTINA CURRAS is an Associate Professor in the Department of Civil and Environmental Engineering. Dr. Curras earned her B.S. in Civil Engineering from the University of California at Davis in 1995, her M.S. in Civil Engineering from the University of California at Davis in 1998, and her Ph.D. from the University of California at Davis in 2000. She has been teaching geotechnical engineering and general engineering topics at the University of Wisconsin-Platteville since then.

Philip Parker, University of Wisconsin-Platteville

PHILIP PARKER, Ph.D., P.E. is an Associate Professor of Civil and Environmental Engineering at UW-Platteville. He received his B.S.., M.S., and Ph.D. in Environmental Engineering from Clarkson University. His primary teaching responsibilities are in the environmental engineering area, but he also enjoys teaching freshman engineering courses and the Computer Applications course. His research interests include solid and hazardous waste management, drinking water purification, and assessing best teaching strategies.

A Homework Problems Database:
Design and Implementation

Abstract
An implementation of a homework problems database is explained. The database allows instructors to categorize problems using several criteria including subject matter, required skill set, difficulty, and the date the problem was last used. The problem solution is also stored reducing the effort to produce an assignment key. The design and implementation of the database is given and assessment of its usefulness is provided.

Background
The homework problems database was developed for CEE2120 (Civil and Environmental Engineering Computer Applications), which is a sophomore-level course required of all civil and environmental engineering majors. Students become proficient using Excel and MathCAD, and are introduced to RISA 2-D, AutoCAD Land Desktop, and Microsoft Access. The course is 3 credits, and students describe the course as one of the most work-intensive yet useful classes that they take. The Fall 2005 Learning Objectives for CEE2120 are provided in the Appendix.

The Need
When teaching students to use computer applications in engineering, several complications arise in the assignment of homework problems. Because the assignments are completed (and often turned in) in electronic form, the opportunity to plagiarize students' work from previous semesters is easier than in most other courses, increasing the temptation to cheat. In order to reduce this temptation, we sought to produce some new homework problems (typically one per assignment) each year and minimize the number of homework problems that are used in consecutive semesters. After several years of teaching, this has led to a vast repository of homework problems. We also had the goal of providing problems in various civil engineering disciplines so students can solve various problems in different technical disciplines. Organizing the problems (which were stored by assignment as word-processing files) and tracking their previous usage became a daunting task. To facilitate our efforts, a database was developed (using Microsoft Access) to store the problems.

The database is now in use and has simplified the creation of homework assignments. The design of this database, its advantages, and usage issues we have encountered will be the focus of the paper. Assessment includes reflection by the instructors who have used the database.

Objectives
The objectives of this project were to create a homework problems database that:

- Organizes more than 300 problems into an easily-retrievable format,
- Allows the instructor to create homework sets efficiently,
- Allows the instructor to create homework sets that are unique from sets created in previous semesters,
- Allows the instructor to create homework sets that focus equally on the various sub-disciplines of civil engineering, and
- Is easily updated.

Database Design

The main table of the homework problems database is the "Problem" table, which is illustrated with its relationships to other tables in Figure 1. The design of the database allows the following:

- Each problem can have an author (linked to a separate author table using the `Author_ID` field) and creation date (`Date_Created` field).
- Some problems are stored in a text format, which can be specified with the `Text_Format` field. Possible formats in our implementation are LaTeX and plain text with the actual problem text stored in the `Problem_Text` field.
- The majority of our problems, however, are in "binary" form (Microsoft Word documents) and these are stored in the `Problem_Binary` field using an OLE Object data type.
- The `Discipline_ID` allows us to assign the problem to a specific engineering discipline (or sub-discipline, e.g., Structural Engineering, Geotechnical Engineering).
- Notes for future reference about the homework problem can be stored in the `Notes` field.
- The instructor can assign a level of difficulty (`Difficulty` field) and the estimated time length required to complete the problem (`Length` field).
- The `Parent_ID` field links to another Problem record, and is a means to track permutations of a problem.
- The problem solution is stored in the `Key` field, which is an OLE Object data type. The `Key` field might contain a spreadsheet page with the solution worked out, for example.
- Each problem also includes a one line summary (`Summary` field) that is useful for reports.

Note that all the relationships with other tables illustrated in Figure 1 are one-to-one. This decreases the flexibility of the database (only one author and one engineering discipline can be specified per problem), but it makes the data simpler to manage.

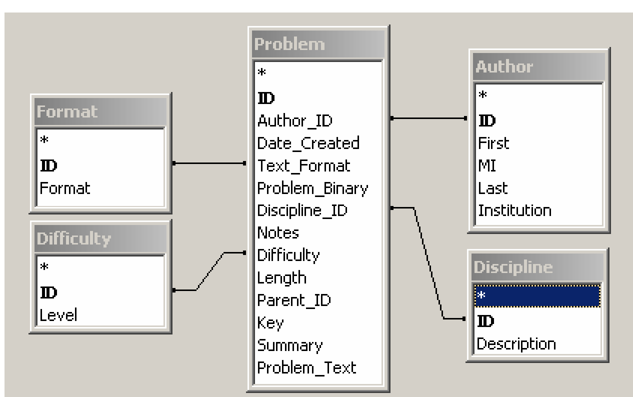

Figure 1 - The problems database table with other "related" tables.

In addition to creating "Problem" records, we also sought to be able to track problem usage by assignment. An obvious advantage to this is the ability to determine how frequently a problem had been used in previous semesters and when it was most recently used. This required a many-to-many relationship (since each assignment could have multiple problems and each problem may have been given in multiple assignments in different semesters). To track problem usage by assignment, a "linking table" was used as illustrated in Figure 2. Another many-to-many relationship was required for the concept(s) covered by a problem. A homework problem will typically have more than one concept (or lesson objective) that is covered. To accomplish this, another linking table was used similar to the one shown in Figure 2.

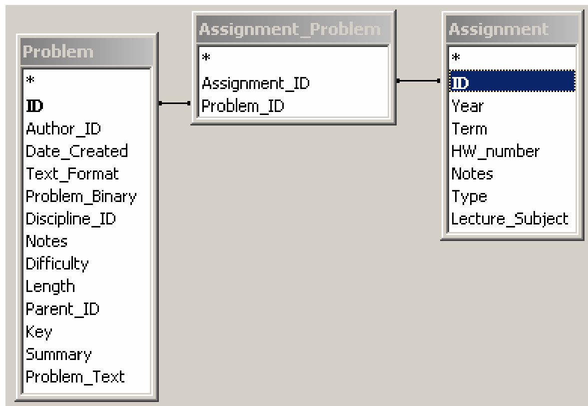

Figure 2 - Creating assignments with problem records.

Database Implementation

Because many different instructors will be using the database, a user interface was created to allow fast access to reports and data-entry forms. The main interface for the database can be seen in Figure 3.

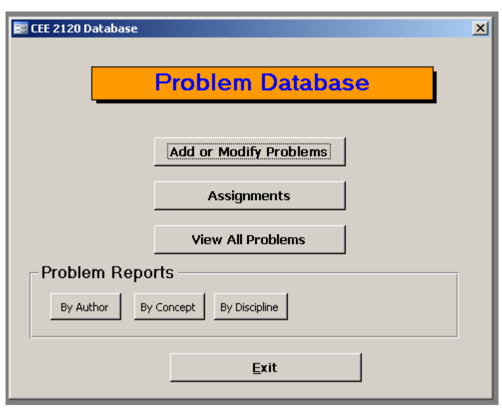

Figure 3 – Main database user interface.

A form was also created to facilitate the creation of new problems. An example is shown in Figure 4 for a problem that is to be solved using Microsoft Excel. Note that:

- Both binary (Microsoft Word) and LaTeX versions of the problem statement are included.
- Multiple problem concepts can be selected.
- Notes allow communication with future instructors.
- The previous years in which the problem was assigned are shown.

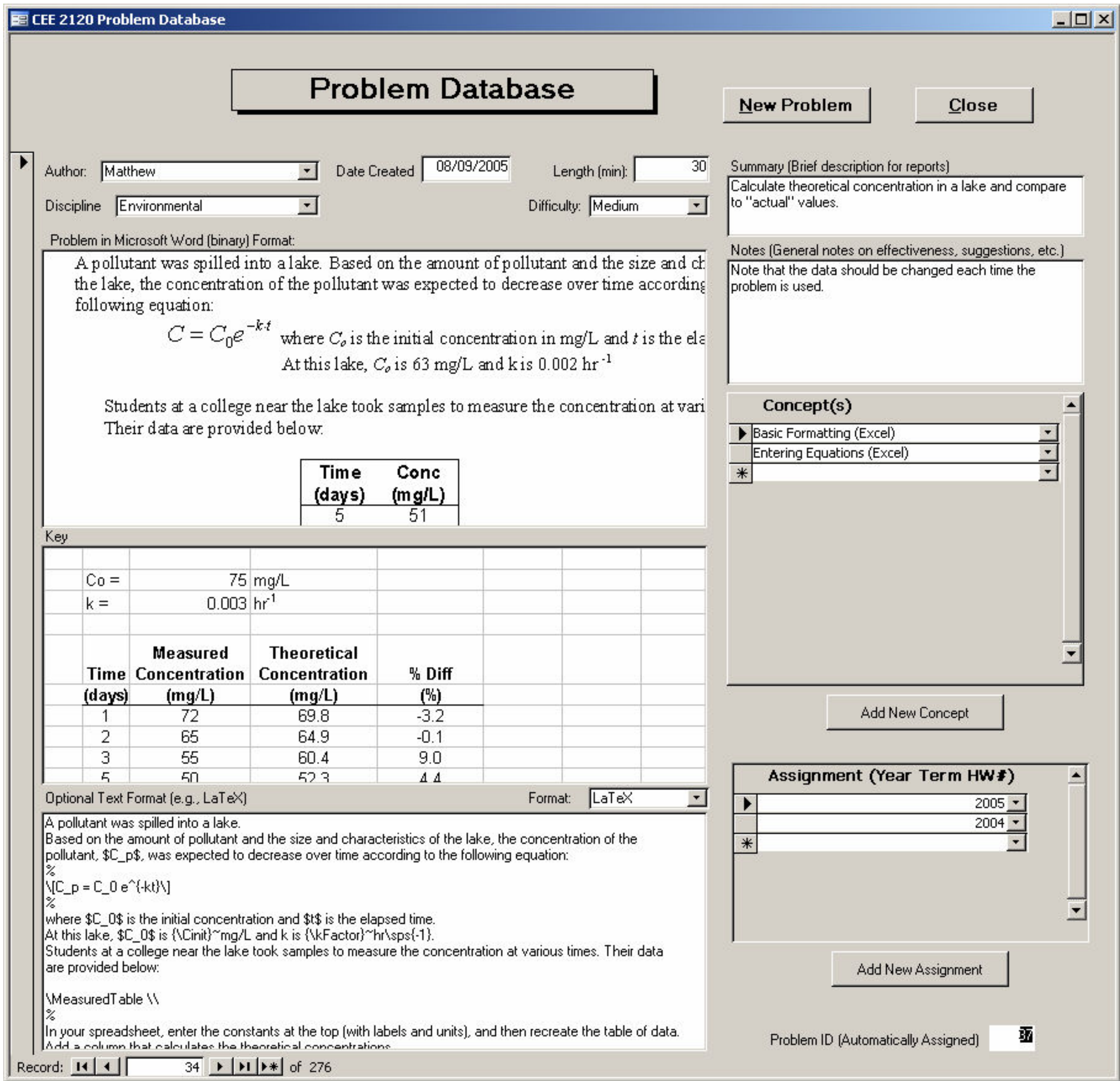

Figure 4 - Form for inputting problem information.

The database was created and populated over Summer 2005, and used for the first time in Fall 2005. At the beginning of the semester, the instructor teaching the course printed out a report summarizing all available problems, sorted by concept, and including the problem number and problem summary. A small portion of this report is shown in Figure 5. When creating assignments throughout the semester, the instructor was able to page through the report to the desired concept(s), find the problem number of appropriate problems, and then enter the database to quickly find and copy the problem statements and solutions. When accessing and copying problems, the instructor could easily see which civil engineering disciplines were being covered, and what the approximate problem lengths and difficulties were, allowing the instructor to easily build suitable assignments. Also, the instructor could determine desired features of new problems.

Problem Summary Listed By Concept

3D Spreadsheets

- [#284] Retrieve stock market information for NYSE and Nasdaq
- [#285] Create summary grading table
- [#193] Steel lookup table with extra credit for using MATCH and INDEX
- [#194] FHWA driver data; vlookup
- [#195] Create a form sheet for an engineering problem
- [#1] Layout the floor plan of the students appartment.

Access Database

- [#325] Exam 2 Part 1 - access query
- [#324] Exam 2 Part 1 - access terms
- [#323] Exam 2 Part 1 - access terms
- [#300] Create a query and a report for a database
- [#299] Create a database
- [#298] Design the layout of a database (describe tables and fields)

Basic Formatting (Excel)

- [#78] introductory problem
- [#1] Layout the floor plan of the students appartment.
- [#266] Import a housing information file and sort.
- [#264] Import and format a file with state populations.
- [#79] Parachute Problem - relative and absolute cell references
- [#261] Find values of some functions in variables w, x, y, and z (Liengme p. 34, #1).

Figure 5 – Segment of problem summary report.

Assessment

The database was useful for accessing existing problems. To keep the database current, however, it is necessary that when an instructor creates a new assignment, the problems used in the assignment are linked to a new assignment record. This linking could be done either when initially copying the problem, or after the entire assignment is built. One disadvantage to this is that if the instructor builds an assignment and then later decides to remove a problem, the database will be corrupted in the sense that the problem will still be linked to the assignment.

The database also added some extra time to the creation of new homework problems. Each new problem must be added to the database. While this was not difficult or particularly time consuming, it did add one more step to creating an assignment.

The organization and flexibility of a database made it ideal for this task. Nevertheless, some of the features of the resultant database did cause some inconvenience. The database, containing over 300 problems and solutions, was over 60 MB. This large size made it unwieldy to use over the shared network drives, requiring the user to store it on a hard drive on the local desktop. When two instructors are teaching and using the database the same semester, this will create a challenge. A client-server application or a web-based database would mitigate this problem.[1]

One feature of Access which caused minor issues is the problem numbering. When using autonumbers in Access, each problem ID is automatically assigned. If a problem is deleted, the ID number is also deleted, resulting in gaps in the ID numbers. However, when using record numbers to find problems, the record numbers do not have gaps. Therefore, ID numbers and record numbers are not the same (see Figure 6), and the difference between the two numbers changes as problems get deleted. This difference can cause some difficulty in finding problems later on, as the reports list the Problem ID, not the record number.

Figure 6 – Problem ID and record number for a particular problem. Note that the record number (34) differs from the Problem ID (37) because previous problems were deleted.

One bug was found in Access which affected the appearance of the forms. When moving from problem to problem, portions of previous problems would frequently remain visible as shown in Figure 7. We have called this failure to clear the form of previous problem information "shadowing." This did not affect the data stored, and the problem text and solution were still assessable, but it could sometimes make viewing the problems difficult.

The database solved the specific problems we initially identified, and made tracking the usage of homework problems much easier. Overall the database did work as intended. Creating varied, appropriate homework assignments was efficient. The problems caused by the database did not outweigh the time saved by using the database.

Also, an automatic assignment generator, in which the instructor could just select the problems and the assignment files would be automatically built, would be extremely useful. Since an automatic assignment generation feature is not available at the current time, it was necessary to copy and paste the problem statements and solutions into new Word, Excel, or MathCad files.

Overall, based on the evaluation of instructors, using the database was an efficient way of creating varied, appropriate homework assignments, leaving the instructor more time to spend making up new problems to fit the gaps left by the selected problems.

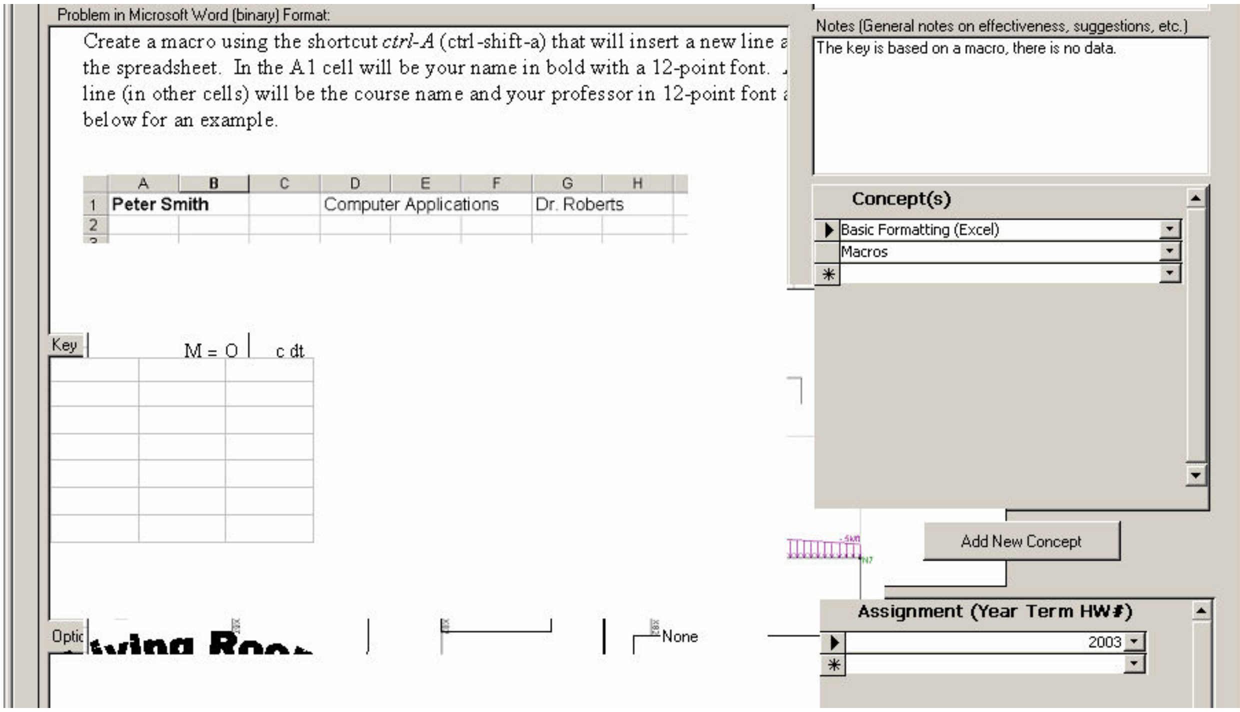

Figure 7 – A particularly egregious example of problem shadowing.

Future Work

Future additions to the database include automatic assignment generation. This would allow an instructor to select the problems for a problem set, generate an assignment sheet for a word processor, and possibly even create the key for the assignment. The user interface will also be improved to allow easier problem creation and database maintenance. A client-server architecture would also help when multiple instructors teach the course in the same semester.

Acknowledgements

This work was supported by a curriculum development grant from the Department of Civil and Environmental Engineering at The University of Wisconsin—Platteville.

References

1. Gehringer, E. F. (2003). "A Database and Search Engine for Sharing Fine-Grained Course Materials over the Web." *Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition*, American Society for Engineering Education, Nashville, TN, June, CD-ROM.

Appendix

Fall 2005 CEE 2120 LEARNING OBJECTIVES

At the end of this semester, you will be able to...

0	◆ List major components of a computer ◆ Explain how a computer's components interrelate ◆ Explain how numbers are stored in a computer and why this can cause slight errors in computations ◆ Use Desire 2 Learn to access class info
1	◆ Review ways to perform basic Office tasks ◆ Use special formatting features of Word, including Equation Editor, bulleted and numbered lists, Draw toolbar, Insert items, tables, Headers and Footers ◆ Use screen capture program
2	◆ Use Windows Explorer to manage files ◆ Create a basic web page using Word ◆ Upload web page to the UWP network
3	◆ Enter equations into Excel ◆ Use "3-D" spreadsheets ◆ Properly format a spreadsheet ◆ Name cells ◆ Print from Excel
4	◆ Correctly and effectively plot data ◆ Format a graph professionally ◆ Print graphs from Excel ◆ Plot multiple series on an x-y graph ◆ Explain differences of x-y and line graphs
5	◆ Plot data using two y-axes ◆ Create and format a bar graph ◆ Plot log-log and semi-log graphs ◆ Parse large data sets into Excel ◆ Sort data ◆ Use range function on data sets ◆ Freeze panes and split screens
6	◆ Write a macro ◆ Edit macros using Visual Basic ◆ Use IF statement in Excel
7	◆ Fit trendlines to data ◆ Explain the "least squares" concept ◆ Model data with trendlines ◆ Choose the best trendline with which to model data

8	◆ Explain what it means to solve an equation ◆ Develop procedures to solve equations ◆ Explain the basis of computer algorithms for solving equations
9	◆ Solve equations using the Newton-Raphson technique ◆ Explain how to use Solver
10	◆ Use Solver to find minima or maxima ◆ Solve engineering problems with Solver ◆ Use "VLOOKUP" command
11	◆ Perform matrix multiplication by hand ◆ Define identity matrix and inverse matrix ◆ Solve simultaneous linear equations by hand using matrix methods
12	◆ Perform matrix algebra in Excel: multiplication, inverse, and transform ◆ Solve simultaneous linear equations in Excel using matrix methods
13	◆ Solve series of equations using Solver ◆ Solve engineering problems involving linear and non-linear series of equations ◆ (Explain the difference between linear and non-linear series of equations) ◆ Use Auditing/tracking tools
14	◆ Evaluate integrals in Excel using trapezoidal and Simpson's rule
15	◆ Create interactive spreadsheet with forms ◆ "Hide" columns and rows ◆ Protect cells and workbooks
16	◆ Explain the basic features of a database ◆ Add a record to a database ◆ Sort and find records in a database
17	◆ Add a table to a database ◆ Set up relationships in a database ◆ Create a database
18	◆ Set up queries in a database ◆ Create a report in a database
19 (2)	◆ Model a truss and frame using RISA. ◆ Find axial forces, shear and moment diagrams, and displacements. ◆ Use load cases in RISA.

20	◆ Evaluate mathematical expressions in Mathcad ◆ Format pages and text in Mathcad ◆ Incorporate units into calculations ◆ Format numeric results ◆ Write equations in terms of variables ◆ Define functions
21	◆ Evaluate derivatives and integrals ◆ Solve for roots using the solve arrow
22	◆ Enter and evaluate matrices ◆ Graph functions and arrays
23	◆ Solve for roots using "root" command and solve blocks ◆ Use solve blocks for minimization and maximization problems
24	◆ Define range variables ◆ Perform numerical integration
25	◆ Fit a linear curve to data ◆ Curve fitting
26	◆ Solve engineering problems with Mathcad
27	◆ Create Land Development (LDD) project ◆ Describe LDD project directory structure ◆ Create an ASCII file of points
28	◆ Import points into LDD ◆ Edit and format points
29	◆ Copy projects from one location to another ◆ Build a surface from points ◆ Create and label contours
30	◆ Use breaklines to modify contours ◆ Create a cross-section of surfaces
31	◆ View surfaces in 3-D using Surface Display ◆ Create "Grading" object ◆ Determine volume of cut and fill
32	◆ Paste one surface onto another ◆ Create a final surface
33	◆ Use a digitizer to create a contour map in LDD