WELDING INFORMATION SYSTEM - DESIGN, OPERATIONS, METHODS

by

VINCENT LATENDRESSE

B.A.Sc. Université de Sherbrooke 1992

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in
THE FACULTY OF GRADUATE STUDIES

(Civil Engineering)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

October 1993

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature) _____

Department of CIVIL ENGINEERING

The University of British Columbia
Vancouver, Canada

Date THURSDAY, 14 OCTOBER 1993

ABSTRACT

An innovative knowledge management system has been developed using the methods and techniques available through micro-computers and WINDOWS 3.1: The **WISDOM** project aims at research and development of an innovative and comprehensive information collection on welding in close cooperation with industry utilizing modern micro-computer techniques of information review, retrieval, storage, updating, and transfer. Emphasis is placed on new and more efficient methods of knowledge communication, pursuing the idea of exploring knowledge (in contrast to rehearing recipes). The development of this project concentrates on the following three main topics.

- Knowledge Base on Steel Metallurgy and Welding;
- Analysis and Design Methods for Welded Steel Connections;
- Design Requirements of Welds.

This includes the considerations of different types of welds and joints and a thorough review of welding history and basic welding procedures.

Table of Content

List of Figures

Acknowledgment

First, I would like to acknowledge the Canadian Welding Bureau who funded and believed in **WISDOM**. Their help was very much appreciated during all my post graduate studies. Also, I would like to thank all of the people who have helped bring this project to this point: Lilian Siu, Howard Nichol, Steve Yee, Jim Greig, Thomas Wong, and especially Andreas Felber.

This undertaking would not have been possible without David Halliday, our industry advisor. His work in the professional partnership program made this venture possible.

I cannot go without saying that Dr. Siegfried F. Stiemer's help, supervision, and advice on the project and many other things essential to the success of this endeavor. I'm grateful and sometimes surprised at the patience he showed towards me.

Finally, I would like to thank my girlfriend, Élise, for her support during the whole time of my studies at U.B.C..

1.0 INTRODUCTION

In this chapter, the basic objectives of the **WISDOM** program will be discussed. The fundamental information that needed to be gathered and be put into an expert system is summarized. A close look is taken at the need in industry and the educational domain for such a software package. The technological explosion that lead to the "birth" of the application is also discussed.

1.1 THE INFORMATION AGE

The term "information age" is well known to most of us, however, the sheer magnitude of its nature is not easily understood. Only in moments when we are buried under a flood of information, be it by periodical magazines or by large data-bases, such as the purchase orders of the last ten years, do we envision what the information age really means. Information can be printed, spoken, and drawn. It can help us to understand a concept from which we can create new methods which again may be put back into the information chain. We definitely need the information and we need even more the skill to manipulate that information.

A word processing program is an important tool for converting information into tangible form. A spreadsheet program is a more transitional tool. It allows one to transform data or formulae into figures or graphs, and permits the extrapolation of data to enable forecasting of events. A drawing program is a mere information handler which deals with the most compact form of information, the picture. An analysis program takes information from an input file through an established calculation procedure to create a new set of information, the response of a structure or process.

Currently, all of these programs act as stand-alone devices and are limited in application and scope. Relational data base programs, on the other hand, exist on a slightly higher plane. They allow data blocks to interact with each other and, better still, permit "links" between individual files and file segments. This type of database management software has been the cornerstone of information management on large computers and has more recently been adapted for use on personal computers.

The **WISDOM** project aims at research and development of an innovative and comprehensive information collection on welding in close cooperation with industry utilizing modern micro-computer techniques of information review, retrieval, storage, updating, and transfer. Emphasis is placed on new and more efficient methods of knowledge communication, pursuing the idea of exploring knowledge (in contrast to rehearsing recipes). The development of this pilot project concentrates on the following three main topics:

1. Background information on steel metallurgy and welding
2. Analysis and design methods for welded connections
3. Code Requirements of Welds

This includes the consideration of different types of welds and joints and a thorough review of welding history and basic welding processes.

1.2 WISDOM OBJECTIVES

Preliminary research on the **WISDOM** project involved consultation with industry to

determine primary objectives. The need for an information system on welding was unquestionably apparent. The content of the pilot project concentrates on basic welding information combined with state of the art design principles and code requirements. The knowledge base is supplemented with graphic images and analysis and design programs. It is hoped that the information in this system will be delivered to the engineer in an efficient and useful manner, so that it might raise welding awareness while promoting efficient welded design. In addition, it is hoped that the **WISDOM** system will close the ever widening gap between the design engineer and the fabricator.

In the most general sense, the primary objective of the **WISDOM** project was to produce an integrated self-paced teaching tool for both engineering professional and student alike. The versatility of the system is maintained by presenting the information in modular form. Screen graphics oriented learning modules encourage the user to become more involved in the learning process than more conventional teaching methods allow. Analysis and design modules promote rapid and efficient connection design in a more consistent and professional manner. A smooth transition from the learning environment to engineering practice is envisioned by providing realistic design tools with a transparent background. In particular, **WISDOM** focuses on three general educational aims which cannot be achieved by conventional textbook techniques:

1. Relations. It is important to facilitate the observation and creation of connections and relationships among ideas, concepts, events, and facts. **WISDOM** provides the software tools that will permit the user to annotate existing modules, create new information, and create new links between existing and upgraded information.

2. Visualization. Wisdom enhances the users ability to visualize and perceive

complex and dynamic phenomena which might otherwise need extensive analytical calculations and graphical or motion picture representations. The full impact of this educational aim may not be realized until further additions are made to the pilot project.

3. Exploration. **WISDOM** encourages exploration of an information-rich environment such that students may discover ideas, themes, and facts on their own.

The following module types are typical of the **WISDOM** pilot project:

1. background welding information modules;
2. specific welding design modules;
3. design modules with code related requirements and design guides;
4. hypertext module with background information.

Besides the above stated educational objectives, **WISDOM** has several technological aims:

1. Reusability. The software industry and the academic research community tolerate by far too much wasteful reinvention of code. A large portion of development time for each of the thousands of software products is spent writing code that performs functions already implemented in many other packages. While this competition is worthwhile for fine tuning special product areas, **WISDOM** uses large portions of commercially available code and remains accessible to future developers.

2. Consistency. An important feature of the **WISDOM** shell is to provide a consistent screen oriented user interface across modules, such that the selection of choices and commands, and the creation and following of links operate identically in all applications. This consistency is extended deep into the coding, such that even data structures and procedures can be easily understood by other developers.

3. Modularity. If a program system like **WISDOM** is developed by more than one person it is mandatory to be not only consistent, but modular. The modularity facilitates partial development and further extensions. Modules have to be created as stand-alone applications and one must by able to test them without interference with other developers. The modules must have the ability to be integrated into the system with little effort.

1.3 TARGET GROUP AND BENEFITS

The main target groups of **WISDOM** project are the practicing engineers and the university students, who wish to specialize in this area. Using such a tool which is based on explorational techniques of education does not require sitting in seminar sessions or attending rigidly scheduled classes.

A survey of representative consulting engineers and fabricators indicated a definite need for an up-to-date, self-paced, learning and application tool for welding. Trying to close the ever widening gap between consultant and fabricator seems to be a timely task. Currently none of the Canadian universities offer an undergraduate civil engineering core course which contains welding as a single specialized topic. It is not expected that this type of material will ever be given at this level, and very little chance exists for an extensive

coverage in the graduate curriculum. However, it would be an excellent opportunity to supply interested students with such a tool for a self-driven exploratory type of studying.

The accessibility and understanding of current knowledge and expertise on welding will be greatly improved by the **WISDOM** project and will lead to more frequent application of welds with a higher degree of confidence. Education and training will be improved, and with this, the general quality of applications in welding. A higher level of confidence in the professional community will further the use of welding.

New areas of research might be identified and intensified. A general raising of the image of welding can be expected. An improved educational preparation plus the continual updating of knowledge for key persons in the decision making process will further the applications of welding of steel structures, improve the quality of design, and increase the efficiency of application.

1.4 DESCRIPTION OF THE WISDOM MODULES

The WISDOM system contains three important modules: WISDOM, workhelp, and the picture library. They interrelate through *hypermedia links*. A user can jump from one component of the software to the next with ease. Novice users will find the operation of the program is facilitated because standard Microsoft WINDOWS commands, buttons, and pull-down menus are used to activate the different functions. Fig 1.1 is a schematic view of how the elements of the system relate.

At the center of the system is the WISDOM index. From this page a user has access to every part of the program. This module is sub system of the main application that contains the leaning modules, the design templates, the electrode list, and the base metal list. These

will be described in more detail further on in the text.

The workhelp sub system divides into the table of content, the topic list, and the topics. These modules contain a large amount of information on welding fundamentals and procedures. Some of the topics are directly linked to the design modules, others have to be accessed with the use of the index or table of content.

The picture library contains figures and diagrams related to the help topics. They illustrate varied information: standard detailing, residual stress diagrams, weld type, etc. This library is linked to the workhelp module and the WISDOM index.

The quick tour guides a new user through the important parts of the software. It describes the use of the basic commands. Also included are copies of the basic types of design templates that have additional operational information attached. The user can start this module from the main index.

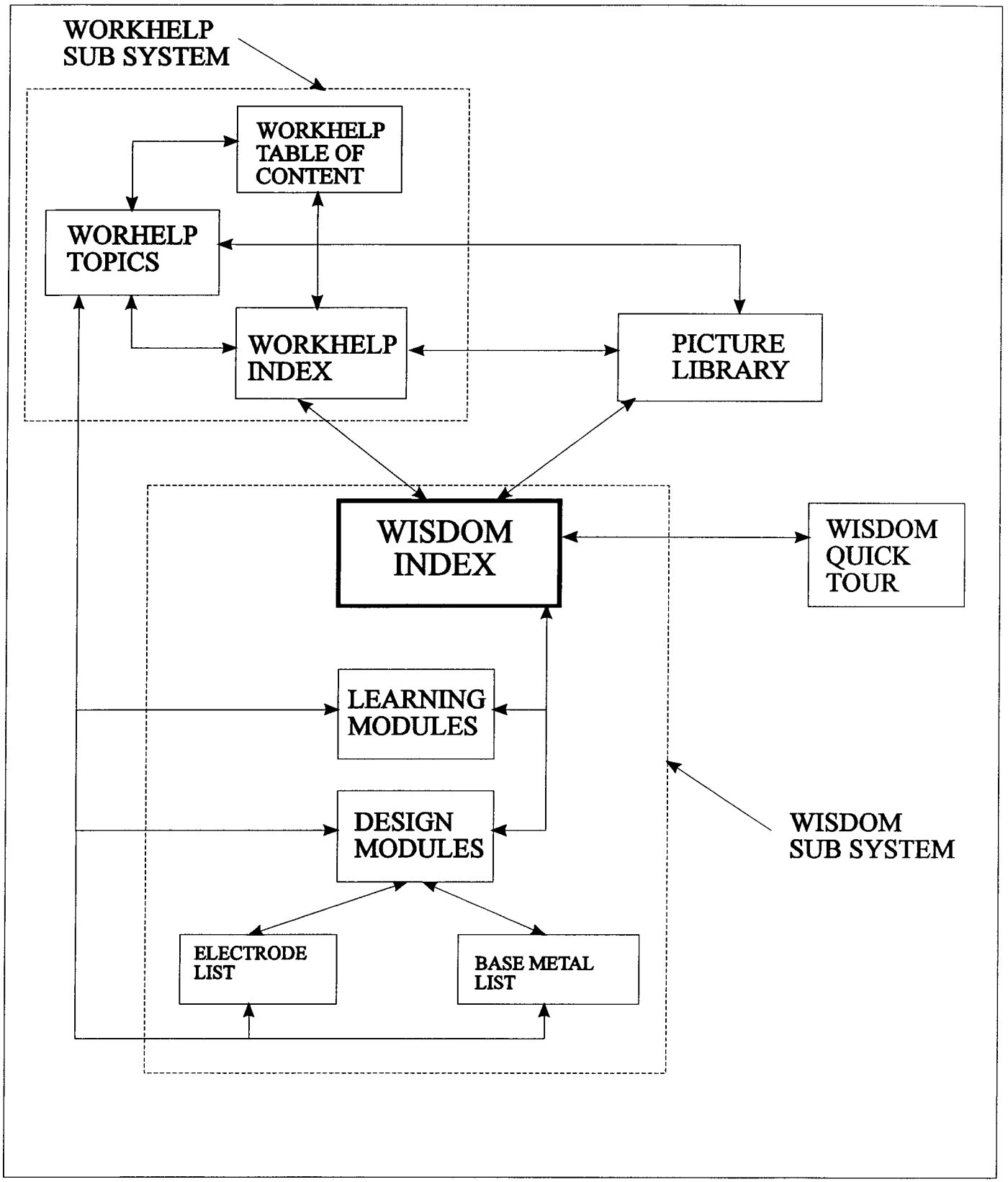

Fig 1.1 Schematic description of WISDOM

2.0 EXPERT AND KNOWLEDGE BASE SYSTEMS

In the following pages, the basic terms used in expert and knowledge base systems definitions are described. The word expert and in its many meanings will be nailed down. The tasks that are to be accomplished by such systems are reviewed.

2.1 WHAT IS AN EXPERT

Bluntly put, the definition of an expert system is a system that does the work of an expert. However, this simple definition brings about two new questions. What is an expert ? What does he do?

Every day life gives us examples. What does an expert do? He takes facts and given values, he analyses them, then he renders a verdict. A medical doctor examines his patients and gives diagnostics. A mechanic listens to stories of how our car broke down and then finds (in principle) the troubles. These two types of experts offer diagnostic help and propose a way to remedy the situation. Other experts, like engineers or lawyers propose solutions to problems with given boundary conditions.

We note from these simple examples the semantic change of the word expert to that of specialist. When we look at a welder or a machinist, we say that he performs specialized work. We might not think of our mechanic as an expert although he is probably a specialist in motor technology.

So in the expression "expert system", the word "expert" has a very broad meaning.

This remark is very important because most people consider experts to be people with vast amounts of knowledge and years of experience in their fields. This definition of the word does not fit exactly in our context. It hides the true meaning of the expression expert system, which is: a program to reproduce the day to day activities of an army of specialists in an organization.

2.2 WHAT DOES AN EXPERT DO?

The word *expert* is used for the remainder of the text, but it will be taken in the much broader sense as defined previously: that of a specialist or a person trained for a precise task.

What does a structural expert do? He asks you what type of building you want, what use it will have, what size and many other things. These are boundary conditions. From these limits, he will emit judgments on the possibilities and the bounds of your project. An engineer uses his knowledge and the limiting factors imposed on him to answer his client's questions.

Of course, it is impossible to know exactly what goes on in the specialist's head. However, we can deduce that with the help of certain facts and natural laws a system is limited to a certain set of solutions. We make the problem smaller by "weeding out" the improbable or impractical answers. The expert uses his acquired experience to do this. This process continues as new facts and limiting factors comes into play, and eventually, and not necessarily in a linear fashion, the problem converges to a final solution.

The classic Newell and Simon artificial intelligence books (Ref.5) discuss this type of reasoning. We refer to it as "*If...Then*" type reasonings. If I see a certain condition then I take this sort of action. Again, we have to take the words' condition and action in a very broad sense. The action can be purely cerebral, like checking a hypothesis or memorizing a new fact.

2.3 EXPERT SYSTEMS AS SYSTEMS BASED ON KNOWLEDGE

It does not matter if an expert is a lawyer or a machinist, he has acquired through his work an expertise (a specialization) which he stores in his head. As seen, looking from the exterior, one can represent information as a series of "*If...Then*" rules. These rules along with situation cues (when to use what), combine to give him the knowledge of the subject. The sum total of this knowledge in computer language translation, stored on a magnetic medium, is called a *knowledge base*. An expert system is therefore a knowledge base system.

One can argue that all computer systems use knowledge. Therefore using information is not the defining characteristic of a knowledge base system. One can characterize these systems by their use of information that is grouped in modular form and quite distinct from the rest of the whole. In an expert system, there are two distinct types of knowledge, the base information (computer language, infrastructure), and the information dealing with the task at hand (the specialist's knowledge).

3.0 IMPLEMENTATION OF AN EXPERT SYSTEM

Although this thesis is not an extensive study of expert systems and their workings, a small discussion of the different parts that make up this type of program will help in understanding the internal workings of **WISDOM**.

To further this understanding, we will discuss the basic building blocks of expert systems and their interactions. Also, the recent advances in technology have permitted the implementation of new systems for information management. The hypertext data arrangement technique is one of these methods and it was used to implement **WISDOM**'s help system. This technique will be defined and discussed.

3.1 ANATOMY OF AN EXPERT SYSTEM

An expert system has three different modules: an inference engine, a user interface, and a database or knowledge base (Fig. 3.1).

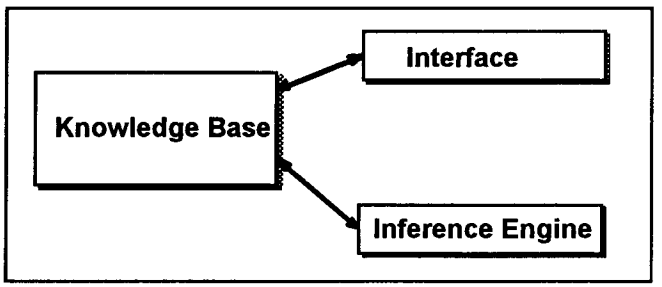

Fig. 3.1 Schematic View of an Expert System.

3.1.1 THE INFERENCE ENGINE

This sub-system controls the access the user has to the knowledge stored in the program. It selects and applies the rules as the user navigates through the different parts of the program.

The program user generates activities that have to be examined by the system, which in turn has to choose a course of action to take (i.e. do nothing, generate a certain message, etc.). It first decides what data is needed to solve the problem at hand. If the data set is incomplete, it goes back to the user or it's database base to complete it. After having assembled the required information, it extracts the rules and restrictions attached to the particular type of problem and executes them. One can implement many strategies of solution finding at this point: forward chaining, backward chaining, etc. Once the program assembles the solution set, it records its conclusion. This step also encompasses a large number of possibilities: printing a simple *yes* or *no* on a computer screen to writing a full situation report. (Fig. 3.2)

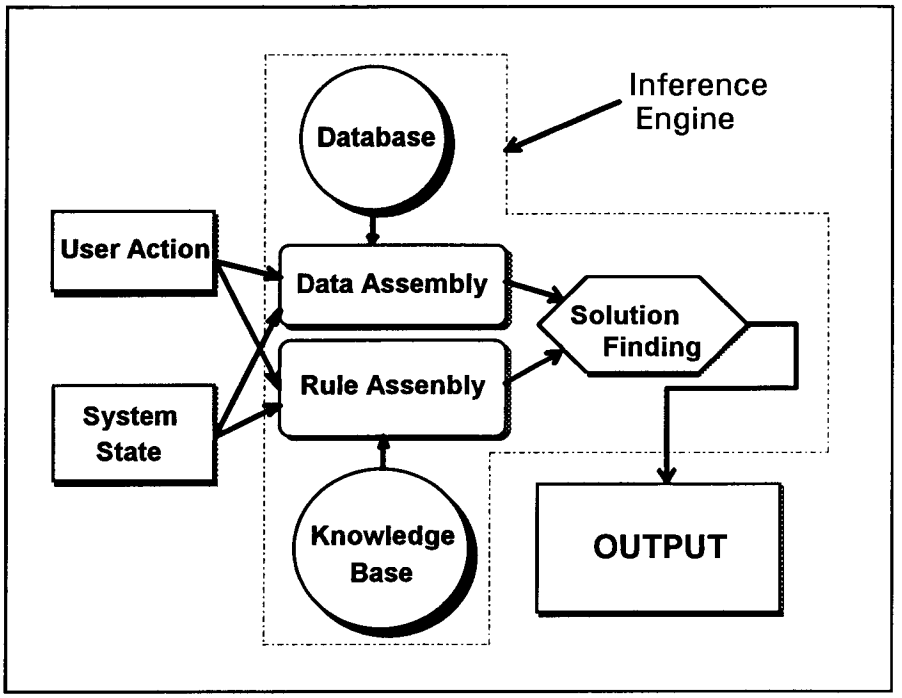

Fig 3.2 Inference Engine.

3.12 INTERFACE

The term *interface* is used to describe the infrastructure that passes the data from the user to the inference engine. This part of the expert system is pivotal to its efficiency. If data entry becomes hard and/or to confusing, a user might become discouraged and just give up. If too much information or too many options are presented at once, the operator might become overwhelmed which reduces his productivity. A good interface should give the user quick access to the most frequently used functions of the program as well as a sense of where he is in the system, and (or) what he is about to do.

Recently, technology has given us a great number of tools to create efficient interfaces. Menu driven interfaces give quick access to functions that are available in the program as

well as laying out these functions in logical groupings so the user reduces his search time. A pointing device, such as a mouse, provides the possibility of making applications more flexible and instinctive to use. Context sensitive help systems provide information on the current system state: which functions can be used, and how to use them at this particular point in the program execution.

Other methods can also help the efficiency of the system. Some examples: a good screen layout will cluster the data at any given time in a small screen space so the user has quick access to it. Minimizing the number of screens that the user has to go through to get the result he wants: this cut the overhead and quickens information access.

3.1.3 KNOWLEDGE BASE

The knowledge base module of an expert system is the entity that contains all the data, facts, rules and constraints that are used to solve a certain problem. This was referred to this as the specialist's knowledge earlier in the text.

In the next sections two types of knowledge base as used in the **WISDOM** program will be discussed. One is more of traditional nature: the design template knowledge base (KB). The other is more open ended: the hypertext knowledge base of the **WISDOM** help system.

3.1.3.1 DESIGN TEMPLATE KNOWLEDGE BASE

In the design templates, the KB replaces the repetitive effort of a specialist's welded connection design. The time it takes to calculate the strength of a connection diminishes

from minutes to seconds. This enables the user to try many options (different sizes of weld beads, different lengths of weld) in a short time.

The type of data, and rules stored in the system consist of "*If...Then*" relations and boundary conditions. Any information retrieved by the inference engine from the knowledge base is usually not visible to the user, only the input values and the results.

3.1.3.2 HELP SYSTEM KNOWLEDGE BASE

The knowledge base in this module uses a more open ended approach to pass on its expertise. The main reason for this change in approach is that the purpose of the help system is quite different from the design template. Whereas the templates try to simulate the repetitive nature of a specialist's work, the help module tries to convey basic information about welding practice and other general subjects. These two sets of knowledge are very different: the former is based on "*If...Then*" rules, the latter on text and picture information. It is evident that one has to use different infrastructures to convey the information. The hypertext support system was chosen to implement the help system.

3.2 HYPERTEXT DEFINITION

The hypertext medium is a relatively novel way to store and manage information: it gives a user the possibility to retrieve data in a non-sequential fashion. The information that most users have experienced in their life is presented in a very linear way: open the book at the first page of chapter one and read on from there. This method has one drawback: Most people don't assimilate information in the same way. An author's view of how to best

organize this document might not be the best solution for another person. If one would provide ways for a user to control the information flow, the speed at which he learns would increase. One of hypertext's great abilities is to provide a quick way to jump from one place in the text to another, so that the reader controls the sequence of knowledge presented to him.

There are also other advantages in using this sort of medium. One can present information in many forms: pictures, sound, video, etc. This multiple medium support system is called multimedia, or hypermedia. All of a sudden, information is not static anymore: the user affects directly his learning path so he feels closely involved in this journey.

In the following sections, the basic concepts of this new technology will be discussed in more detail.

3.2.1 HYPERTEXT SYSTEM

The concepts of hypertext (or hypermedia) where laid out as early as 1945 by researchers like Vannevar Bush (Ref.11). Bush, who was at that time director of the US Government's Office of Scientific Research and Development, wrote an article in the Atlantic Monthly in which he sketches his vision as a tool that will aid knowledge workers. It would retrieve and organize vast amounts of information and provide ways of retrieving selected parts of this "database" very quickly. He even proposed the concepts of individual links that transfer the user from one point to another in the database. This vision of information storage was not realized at this time because the microfilm technology was not efficient enough. Nevertheless, these ideas became the founding concepts of hypertext.

3.2.2 HYPERTEXT BASICS

There are three basic features to a hypertext/hypermedia knowledge base:

- A network of nodes,
- Links between nodes,
- Buttons.

Nodes are the parts of the hypertext that contain the text and other kind of information (picture, sound, video). It's the information the user is interested in. The links connect the nodes together: they permit the user to jump from subject to subject. Some examples of links are tables of content, index to documents, etc. Finally buttons are specific locations in the hypertext where a user can jump from one node to another via a link. Basically, these are the physical representations of the links in the hypertext. (Fig.3.3)

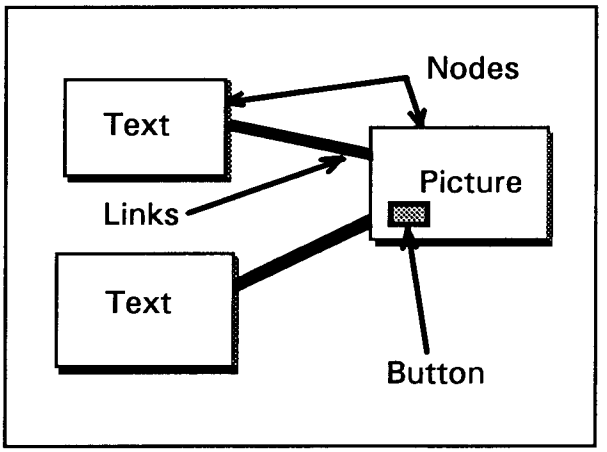

Fig 3.3 Hypertext representation.

3.2.3 ADVANTAGES OF HYPERTEXT

As we have said before, hypertext enables the reader to make decisions on how to sequence the available information. It does not restrict the user to some predetermined path to an answer because of the existence of multiple links to every node. These types of links between related topics help the user to learn by association. Parallels between subjects that were not obvious in a more traditional text become very apparent in this type of structure. The knowledge is also stored in many formats so that the user has many ways of understanding the same idea.

Some of the advantages of using a hypertext system can be listed as follows:

- Rapid browsing,
- Non-linear organization, multiple paths to information,
- Promotes associative thinking,
- Multi media capabilities.

3.2.3.1 RAPID BROWSING

Hypertext permits the user to rapidly switch from subject to subject. He/she can navigate through the knowledge base to get the data he/she needs in a very efficient manner. Also, the reader can switch from problem solving to information gathering in a fraction of a second. Additionally, there is a possibility of context sensitive learning: if the user needs information on the specific problem he is dealing with, the hypertext system can bring him to the pertinent subject matter quickly.

3.2.3.2 NON LINEAR ORGANIZATION, MULTIPLE PATHS TO INFORMATION

As was discussed earlier, information in a hypertext system is not stored in a predetermined way. A user can access what he finds important in many different ways. The programmer can define a more predetermined structure in a table of content or an index of topics. Also even "hard wired" training sessions can be implemented, but in general, the user is quite free to do as he/she pleases with the information flow.

3.2.3.3 PROMOTES ASSOCIATIVE THINKING

The use of a hypertext system promotes associative thinking. The reader cannot go through the hypertext passively. When he finishes exploring the information at a node he has to take action or go to another subject. By jumping from node to node, he creates associations in his head that help him get a deeper sense of the subject matter. This is probably due to the fact that the humain brain does not store information in a sequential way but by associations of ideas. So hypertext provides a more natural way of understanding, remembering, and learning concepts.

3.2.3.4 MULTI MEDIA CAPABILITIES

As hypertext is not limited to textual information, it has the capability of informing the user in many ways. Pictures, graphs, sound, video can all be used to illustrate principles. Thus, the user can grasp the concepts that are difficult to describe on paper by viewing a short video clip, for example. The user has many chances of approaching the knowledge: he/she can choose the information delivery medium which works best for him/her.

3.2.4 DISADVANTAGES OF HYPERTEXT

There are many reasons for using a hypertext system as noted earlier: rapid browsing, non linear organization, etc. However, there are some problems using a hypertext/hypermedia system:

- Information can be difficult to find,
- Users can become lost in hyperspace,
- There can be difficulties in maintaining a hypertext system.

3.2.4.1 INFORMATION IS DIFFICULT TO FIND

Usually, most information stored in a hypertext system is easily accessible. Nevertheless, as the number of topics in the knowledge base grows, so does the number of links it contains. There is a point where a user may reach a state of learner's block. It might be very hard to decide where to go next because of the great number of possibilities offered.

A programmer could choose to restrict and organize the information that is available to a user. Cues and signals to a user should follow a certain convention, so that no information becomes hidden or lost.

3.2.4.2 LOST IN HYPERSPACE

The knowledge bases can become complex. A user can sometime become very confused as to where he is in the system. Where did he come from? Where is he going? A number of factors can contribute to this state of affairs: large number of links, uncertainty of the destination, inexperience of the user, etc.

A programmer can implement a number of solutions to this problem: he/she can furnish a map of the system (links and nodes), make a list of the subjects the user consults so he can backtrack with ease. These methods solve most of the ambiguities but they take time to implement, thereby costing effort and money.

3.2.4.3 DIFFICULTY IN MAINTAINING

A significant amout of effort goes into creating a hypertext/hypermedia network and into maintaining it. This stems from the fact that whenever we add links or nodes, existing ones become more or less important or even obsolete. Much work has to go into pruning and upkeeping the network. One cannot do this haphazardly: specialists that are aware of the implication of their action have to work on the project. Needless to say, it takes much energy and time.

4.0 THE WISDOM PROGRAM

The previous chapters reviewed basic techniques developed to manage information. With today's high speed computers, the implementation of knowledge systems is now possible. The welding domain has been a very specialized area in which many engineers have only limited experience. For this reason, it was a very good candidate for casting as an expert system. The following pages describe the different modules of **WISDOM,** the welding expert system. At first, the main module is described, it contains all the design modules. The basic screen layouts, different analysis procedures and interactive navigation commands are reviewed. The final part of this chapter looks at the knowledge base components and the interactions between them and the main calculation module.

4.1 WISDOM INDEX

The first screen, containing the index can be used to navigate to all the general modules of the **WISDOM** system. The user does this by clicking on the appropriate button. Currently, **WISDOM** contains three operating modules: the design modules, the knowledge base and the figure library. All of these are described in more detail in the following text. An introductory tour and other sub programs loosely related to the **WISDOM** system are also accessible by the buttons on this page.

4.2 THE DESIGN MODULES

When one clicks the *"to design module..." button*, it branches to the template index. This screen is divided into three main categories: the elastic and plastic weld group design and the design of connections. Here again, the user can click on the appropriate button to access the template of his choice. The elastic and plastic weld group design templates and

its formulae replace the equivalent tables from the CISC "Handbook of Steel Construction" (Ref.6). A short summary of these two types of design methods and a short description of the use of the three types of design modules used in **WISDOM** follows.

4.2.1 ELASTIC ANALYSIS

Elastic analysis is a proven and safe way of calculating the strength of eccentrically welded connections. One has to break down the continuous beads, that make up the weld group, into many small elements of length Dl for the analysis. All of these small pieces contribute differently to the overall resistance of the weld group.

For eccentrically loaded connections, it is easier to consider the factored shear per unit length of weld; which is also called the factored shear flow (qf). It is the resultant produced by the concentric shear load and the torsional moment acting on the connection.

Four basic hypotheses are made in elastic analysis.

a) The concentric shear load is equally distributed on all the weld line elements (elements of length Dl).

b) A linear strain distribution is assumed, that is, the resulting force couple submits a unitary element to a force proportional to the radial distance of this element to the center of gravity of the group. The deformation occurs in a direction that is perpendicular to this radius line.

c) The shear flow developed in the weld group due to the concentric shear force and the torsional moment at the centroid, sum vectorially to produce a resulting shear flow, which typically varies along the length of the weld.

d) The factored resistance of the weld group is reached when the resultant shear flow in the most critical region reaches the factored critical shear flow of an unitary element.

The following equations result from the preceding hypotheses:

$$qf = \sqrt{\left(\frac{P}{l} + \frac{Mxm}{R}\right)^2 + \left(\frac{H}{l} + \frac{Mym}{R}\right)^2} \leq Dqr \qquad (1)$$

The nominal size of the fillet weld is given by:

$$D \geq \frac{qf}{qr} \qquad (2)$$

Figure 4.1, taken from Picard & Beaulieu's "Calcul aux états limite des charpentes d'acier" (Ref. 9), describes the terms used in the equation.

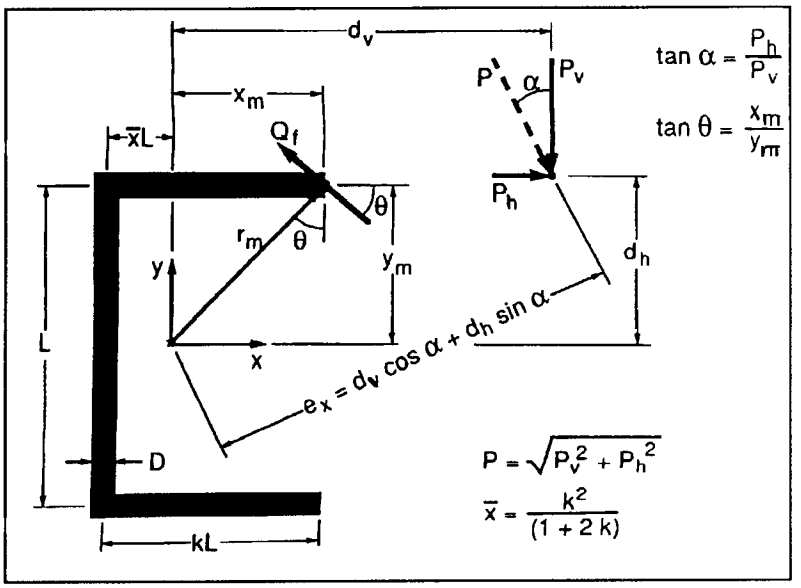

Fig 4.1 Elastic analysis

In equation (1), we use the following notation:

l = total length of weld = $\int dl$;

Xm = abscissa of the critical weld point taken from the center of gravity;

Ym = ordinate of the critical weld point taken from the center of gravity;

r = distance of any weld element from the center of gravity;

R = Geometric Constant of the weld group = $\int r^2 dl$.

If a weld group has beads parallel to the X and Y axis, then we calculate the parameter R with this formula.

$$R = \sum_{i=1}^{n} (I_{xi} + I_{yi}) \qquad (3)$$

The parameters are defined as:

n = number of weld beads;

I_{xi} = Second moment of inertia about the X axis of the weld group, per millimeter of nominal thickness (mm^3);

I_{yi} = Second moment of inertia about the Y axis of the weld group, per millimeter of nominal thickness (mm^3).

The elastic design templates of **WISDOM** use these basic principles. The most critical area for each of the weld group geometries is determined. These areas are found by taking into consideration both their positions relative to the center of gravity and the orientation of the forces applied on the weld group. The maximum factored shear flow can be found at these points. Thereafter, the factored shear resistance at that point is compared to the weld resistance to evaluate if the weld bead size is adequate.

The **WISDOM** program contains a few elastic design templates for different design cases. They all support both horizontal and vertical loads or a combination of the two. The electrode type is also an input parameter. The following geometric layouts are available:

- Box shaped weld group,
- C shaped weld group,
- L shaped weld group,

- Two parallel vertical weld lines,
- Two parallel horizontal weld lines.

4.2.2 THE ELASTIC DESIGN TEMPLATE

The screen of this type of module is separated into two main parts: the parameter input region and calculation region. (Fig 4.2)

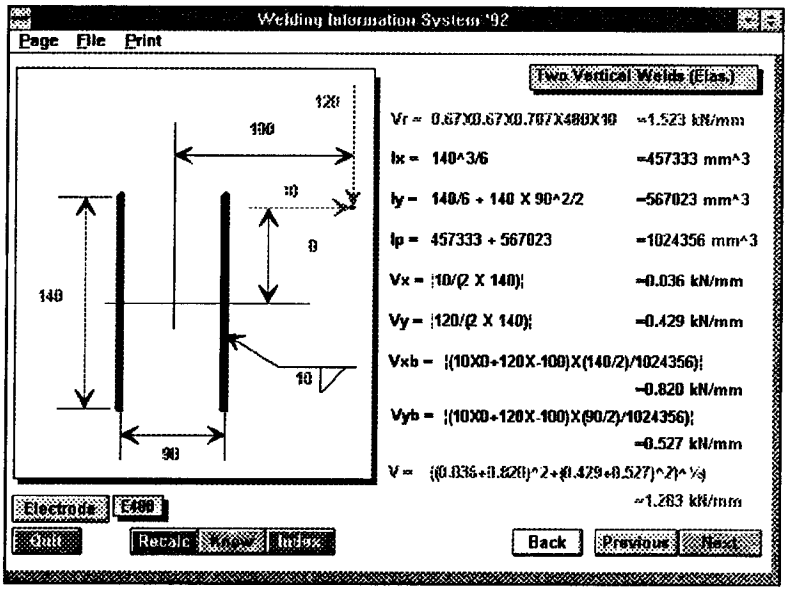

Fig. 4.2 Elastic design template.

One can enter all the parameters needed to complete the calculations with the help of a "active drawing" describing the weld group geometry. Active drawings are figures containing alterable input fields. The physical dimensions like the length and size of the weld beads are needed, as well as the horizontal and vertical loads with their eccentricities from the weld group center of gravity. If there is no horizontal load, for example, one needs to enter "0" in the appropriate input location.

One can change the electrode type by pressing the *electrode button* situated in the bottom left corner of the screen. A small text field to the left of the button lists the current choice of electrode. If the control activates, a small pop-up window appears, (Fig. 4.3). It contains a list of all the major types of electrodes. Other reference information included is the ultimate weld strength in MPa and the associated base metal grade. The current choice in this window highlights in yellow; this highlight color will toggle as the user makes new choices. When the changes are final, the user can press the *OK button* to return to the design module.

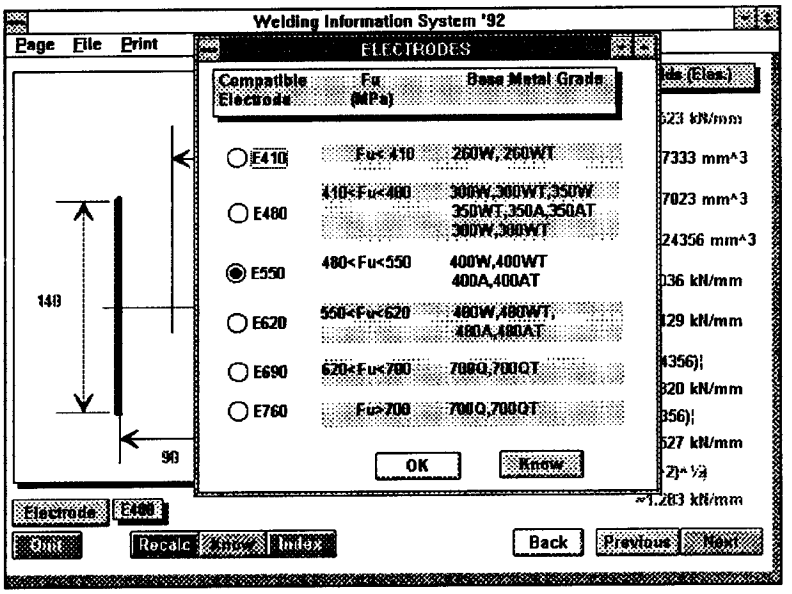

Fig. 4.3 Electrode pop-up window.

One enters the different values, the overall strength of the weld group is recalculated automatically in the right part of the screen. The user can force the recalculation of the equations by pressing the *Recalc button* on the navigation bar at the bottom of the template. The overall resistance value (Vr) in kN/mm uses a highlight color as well as the unit length shear flow value(V). When the highlighting color is blue the resistance of the

group is sufficient. If the color is red, the resistance is smaller than the flow, which means that the design is not adequate.

One of the templates has a special feature. The angle weld module has an extra button situated next to the *electrode button*. It controls the change of geometry of the weld group (Fig. 4.4). The use of this control is quite similar to that of its neighbours. When pressed, a pop-up window containing the four available choices appears (Fig. 4.5). The user just has to click on the options that suits him and press the *OK button* to return to the main program. He/she should perform this operation before entering the weld group dimensions since the four have independent calculation modules.

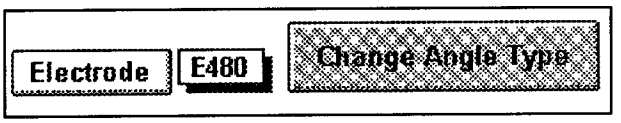

Fig. 4.4 L shaped weld.

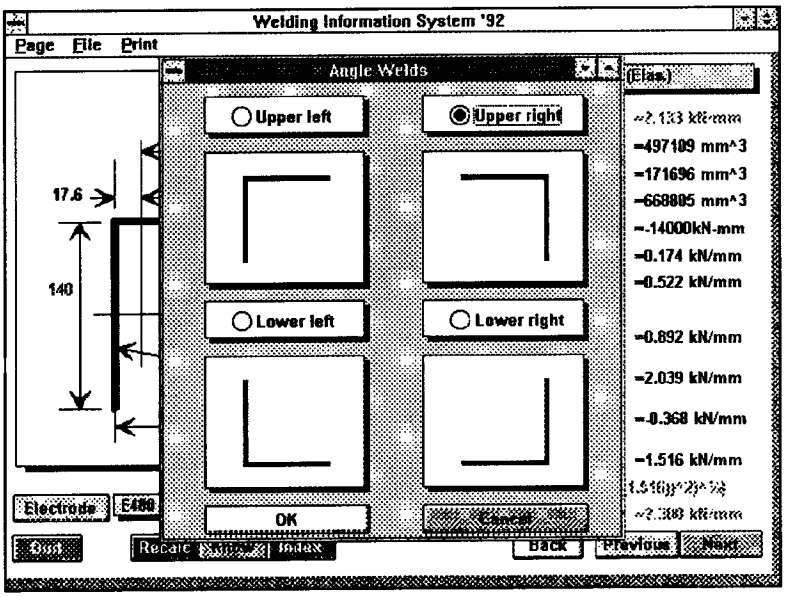

Fig. 4.5 Pop up to change the L shape geometry.

4.2.3 ULTIMATE CAPACITY ANALYSIS (PLASTIC)

The elastic analysis assumes that the weld group will behave elastically until its ultimate capacity is reached. This hypothesis is wrong. Firstly, the weld's force-deformation curve is not linear. Secondly, the welded material around the beads plastifies due to high local stresses, causing the connection as a whole to deform in a highly non-linear way even in light loading conditions. Therefore, one assumes certain conservative hypotheses to use this method of calculation. The full strength of the weld group is not taken into account, which in return reduces the efficiency of a weld analysed by this method.

As in elastic analysis, the weld beads are subdivided into small elements of length Dl. The analysis procedure takes into account the angle between the direction of the force and the weld bead since the resistance of the weld elements varies with this angle (θ_i). This element then has his own load deformation curve characterized by the ultimate level (Δ_u) from which one can calculate its critical shear flow (Fig. 4.6). To define these two parameters, we use equations (4) and (5) are used which were developed by Butler *et al.* (Ref.10). The resistance of the weld group is calculated by summing the individual element's capacities.

$$\Delta u = 0.225\ (\theta_i + 5)^{-0.47} \tag{4}$$

$$Qu = \frac{10 + \theta}{0.92 + 0.0603\,\theta} \tag{5}$$

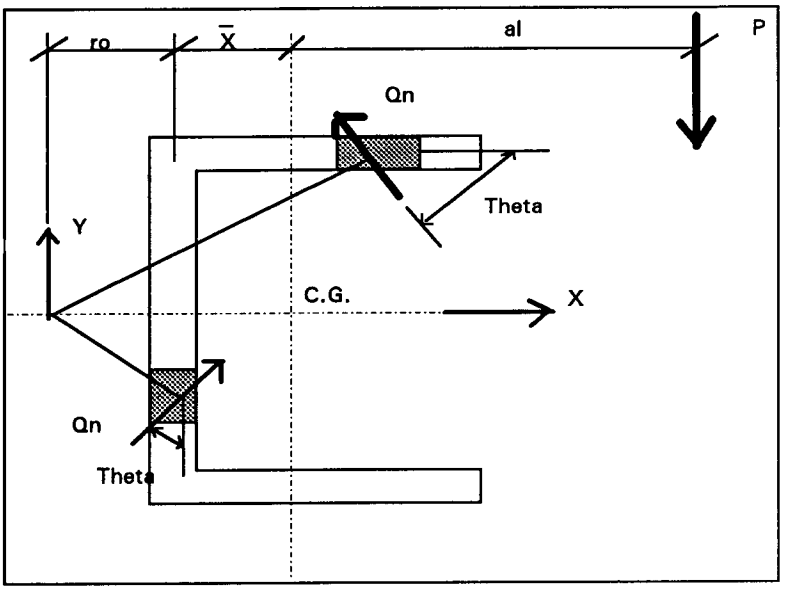

Fig. 4.6 Ultimate Capacity Analysis

These two parameters only describe the last point on the load-deformation curve. It is obvious that not all the weld elements will reach ultimate deformation at the same time. So, an equation has to be defined to calculate the shear flow in an individual element (Qi) for a given deformation level (Δi) (equation 6).

$$Qu = Qu(1-e^{-\lambda i \Delta i})^{\lambda i} \tag{6}$$

where

$$\mu i = 2.95\, e^{0.0114\, \theta i} \tag{7}$$

and

$$\lambda i = 0{,}40\, e^{0{,}0146\, \theta i} \tag{8}$$

The prediction of the ultimate connection strength is based on the following assumptions:

a) The weld group rotates around the instantaneous center of rotation. This center of rotation is distinct from the center of gravity of the group. Its position is determined from equilibrium considerations instead of using geometry.

b) The direction of the resulting force on a weld element is perpendicular to the line joining the center of rotation and center of the element.

c) The deformation of a weld element is directly proportional to the distance between that element and the center of rotation. This deformation occurs in a direction perpendicular to the radius line that connects the center of rotation and the weld element.

d) The ultimate capacity of the connection is reached when the most solicited weld element attains its ultimate deformation level.

This method of calculation uses values that are presented in tables of the CISC's "Handbook of Steel Construction" (Ref. 6). If the specific geometry of the weld group does not exactly fit table input parameters, interpolation was used between known values to calculate the "C" coefficient. This "C" coefficient multiplied with the nominal size of the beads and their total length, gives the ultimate strength of a weld group.

The **WISDOM** program however uses an algorithm based on the equations of statics and the non-linear load-deformation curve to calculate the "C" factor for a given geometry. It does not interpolate between any table values. This method eliminates calculation errors and makes the use of plastic design more versatile and generally easier.

The plastic calculation templates are limited to vertical loads and the moment only, although they support the use of all electrode types. The geometric layouts supported by **WISDOM** are the following:

- Box shaped weld groups,
- C shaped weld groups (loads on the open or closed sides),
- L shaped weld groups (loads on the open or closed sides),
- Two parallel horizontal weld lines,
- Two parallel horizontal weld lines.

4.2.4 THE PLASTIC DESIGN MODULES

The plastic design module screen is divided into three main parts: the graphic input section, the calculation section and result summary.

The dimensions of the weld group and the applied loads are entered after which an external C language dynamic link library calculates the instantaneous center of rotation associated with these input values. The library's output is the "C" coefficient: the equivalent of the variable of the same name used in the CISC design table (Ref. 6). Then we can calculate the associated values like the resistance are calculated using this parameter.

The output or calculation section lists all these values and their associated formulae. The software does not calculate the "C" value with the formula shown, the external program

does that with an iteration process. The formula is shown that way in order to ease the user's comprehension.

At the bottom of the screen, the result section shows graphically the relation between the resistance and the loads. Both color and symbols are used in this template to communicate the adequacy of the strength of the weld group. If the color turns to red, the load is too high,but when it turns to blue, the resistance is adequate.

4.2.5 THE CONNECTION DESIGN MODULES

At this point,the design of three types of basic steel connections has been implemented: a simple tension member, a double angle connection, and a flexible beam to column (B-C) connection. The tension member and double angle templates will be discussed simultaneously because of their similar workings, the description of the flexible B-C connection will come last.

4.2.5.1 TENSION MEMBER AND DOUBLE BRACKET CONNECTION

The general format used in these templates is quite similar to the one described previously. The main screen is divided into two parts: input fields for the geometry and result area (Fig. 4.7).

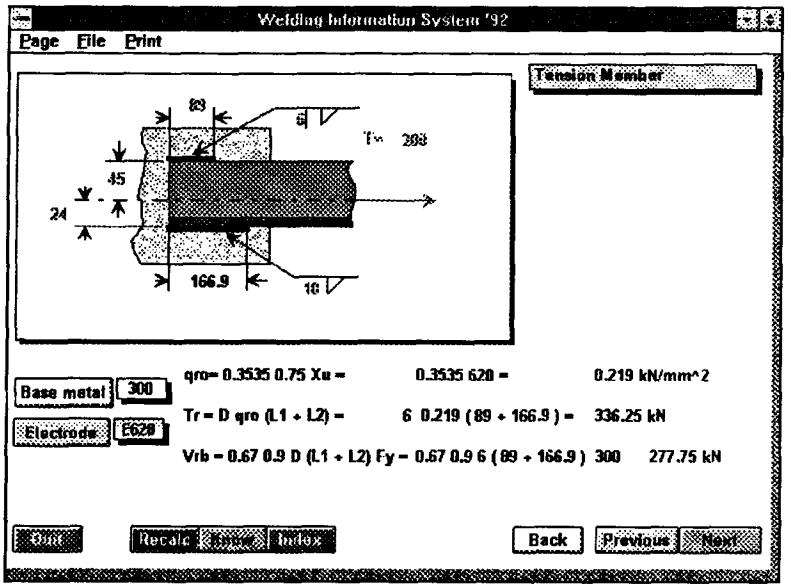

Fig. 4.7 Tension member design template.

The geometry of the whole assembly is entered in the appropriate fields situated in the top left corner of the screen. The program executes the calculations every time the user changes fields or presses the enter key. The electrode and base metal types are variables in this problem; the user can change their values using the buttons situated just above the navigation bar on the left side of the screen. Both buttons have small listings of the current valves used in the calculations. When the user clicks the base metal control, for example, a pop up window appears. It shows the most common steel grades (Fig. 4.8). Once the user has chosen the appropriate type for his problem, he can click on the *OK button* to return to the main module.

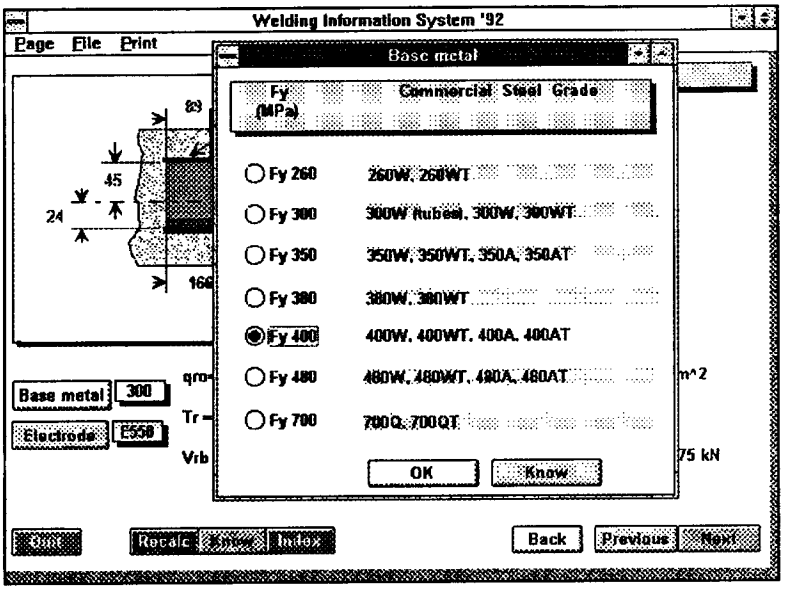

Fig. 4.8 Commercial steel grade pop up window.

Another feature of these two modules is the graphical reprensentation of failure or adequacy: When the resistance of the overall arrangement is lower then the applied load the picture of the connection changes to show the failed state (Figs. 4.9 and 4.10).

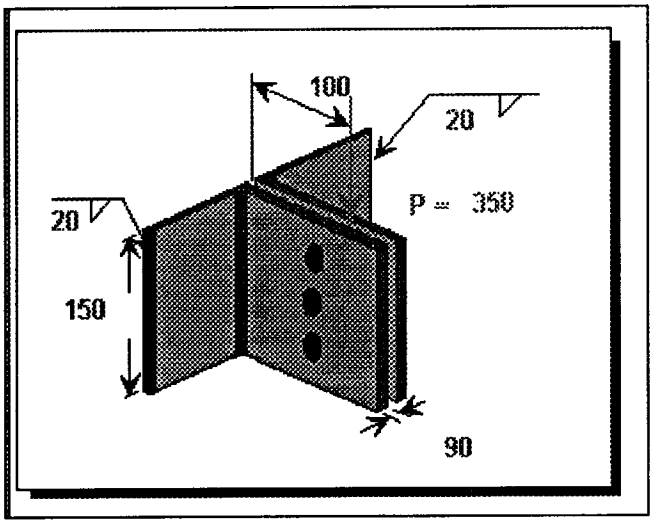

Fig. 4.9 Resistance OK.

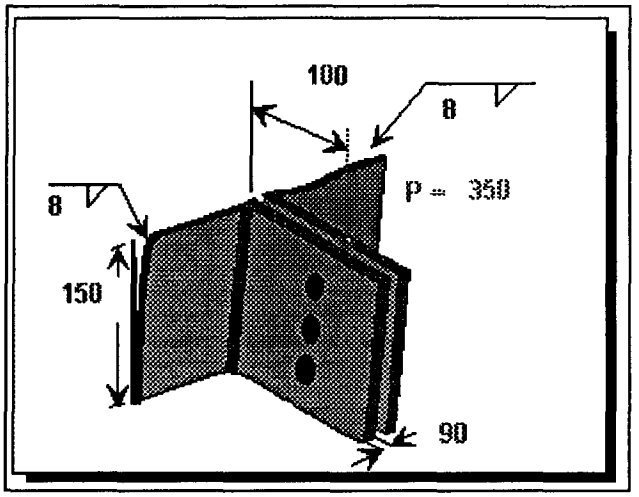

Fig 4.10 Resistance not OK.

It should be noted, that the program does not currently check the minimum and maximum permissible weld bead sizes. The user should still refer to the proper tables in the CISC "Handbook of Steel Construction" (Ref. 6) or any other suitable text to check the weld bead size against the plate thicknesses.

4.2.5.2 FLEXIBLE BEAM TO COLUMN CONNECTION

This design module can be used to replace a part of the "Seated Beam Shear Connections" tables used in the CISC "Handbook of Steel Construction" (Ref. 6). This connection consist of a thick angle welded to column flange. Most of the load is transferred to the column only in shear, transfer of moments to the support is neglected. For further information about this type of connection refer to Ref. 6. The plastic design approach has been chosen for the analysis. Contrary to most other design templates, the results are not listed on the same screen as the input parameters. This approach was used because the number of entries was too large to be contained in a small area. (Fig. 4.11 Fig.4.12)

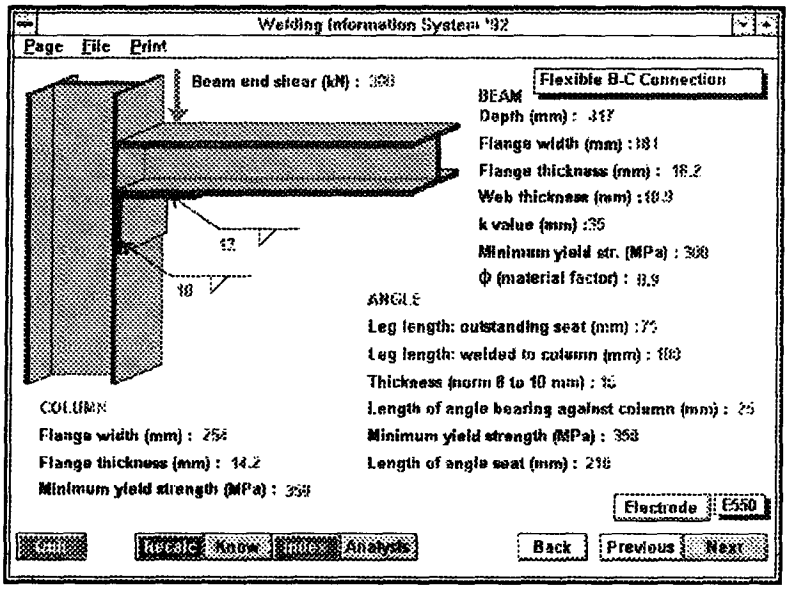

Fig. 4.11 Flexible beam to column design template.

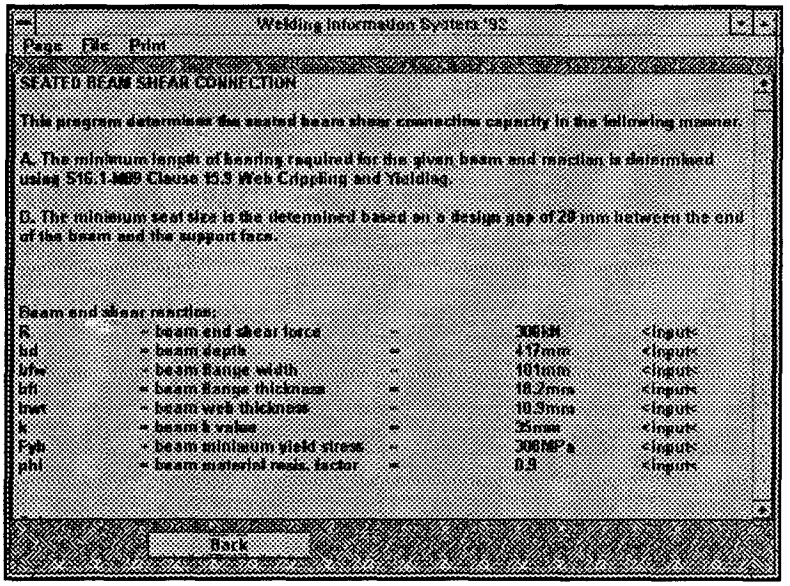

Fig. 4.12 Analysis Page.

The input values are separated into four distinct regions. Each section has been reserved for one of the assembly's parts: the column, the beam, the angle, and weld. Most of the entries are self-explanatory and a small pictorial is provided in the help system. If the user places the cursor over any of the feature of the connection (column, beam or angle), the mouse pointer changes to the help pointer (Fig. 4.13). Then if he clicks the right mouse button, a small diagram appears. It shows the part of interest with its descriptive terms (Fig.4.14). Once the user finishes the query, he can click on the *OK button* to return.

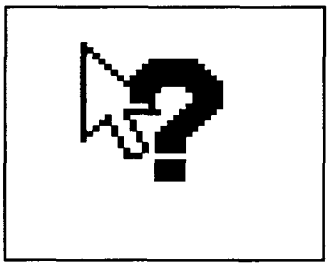

Fig 4.13 Help cursor.

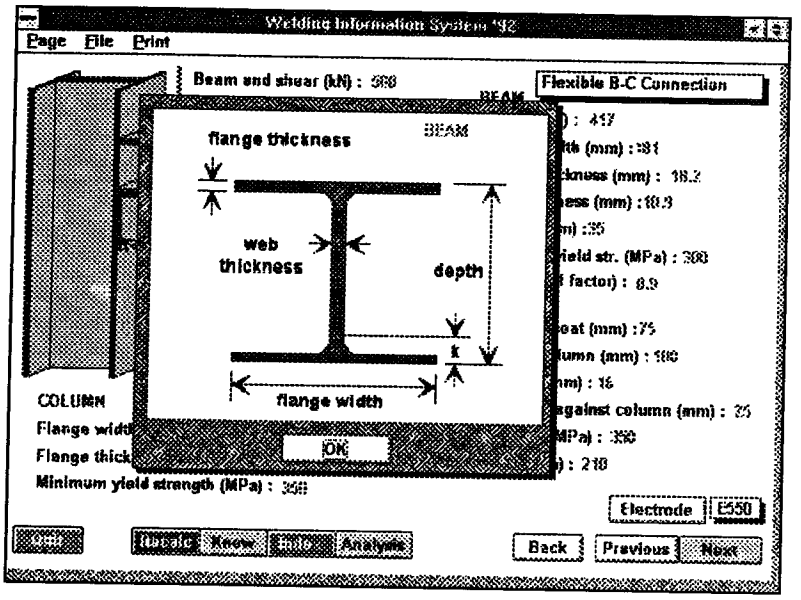

Fig.4.14 Help pop up window.

The type of electrode and base metal type is also of interest in this problem. One can change the electrode type by using the same operating steps described previously: the user clicks on the button in the right corner of the screen, chooses the appropriate type and then clicks on "OK." As for the base metal type, this template uses a slightly different entry method. The user has to place the mouse cursor over the words "minimum yield strength (MPa)" on the screen (the mouse cursor changes its shape (Fig. 4.15)) and then press the right mouse button to make the base metal window appear. The user can input the rest of the values directly through the use of the keyboard.

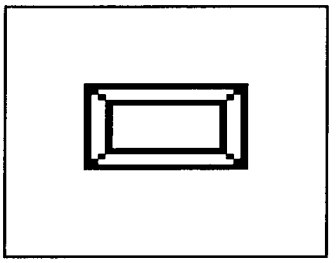

Fig. 4.15 Box cursor.

Once the operator has replaced all the design parameters, he can click on the *analysis button*, situated in this module only next to the *index button* on the *navigation bar*, to get the results. (Fig. 4.12) The program shows a number of intermediate values: this gives an indication of possible design improvements. Clicking on the *back button* returns the input page to permit corrections to be made.

4.3 THE NAVIGATION BAR

A few "standard" buttons are used throughout the WISDOM system. The user will find them at the bottom of the page for ease of access in the form of a typical navigation bar (Fig.4.16). The following text describes the most common buttons in detail.

Fig. 4.16 Typical Navigation Bar.

4.3.1 THE QUIT BUTTON

This button allows the user to quit the current module at any time. If the current module is the picture library, the knowledge base takes back the control. If the current module is the knowledge base, the main design module takes the control.

4.3.2 THE RECALL BUTTON

The *recalculation button* will initiate the recalculation of the current design module. Usually, the user lets the program do the recalculations automatically by changing the input values or by pressing the *enter key* on the keyboard. However, this feature has a "fail-safe" way to ensure that the calculations on the screen represent the results calculated from the current geometric parameters.

4.3.3 THE KNOW BUTTON

The *know button* is the physical representation of one of the most important features of the WISDOM system. It provides a context sensitive link between the current module and the knowledge base. For example, if one clicks on the *know button* situated in the double angle connection module (Fig. 4.17) it will take you to the related topic of the knowledge base (Fig. 4.18). (See the knowledge base section for more information). If no subject link exists, the *know button* will leave the user in the index part of hypertext system.

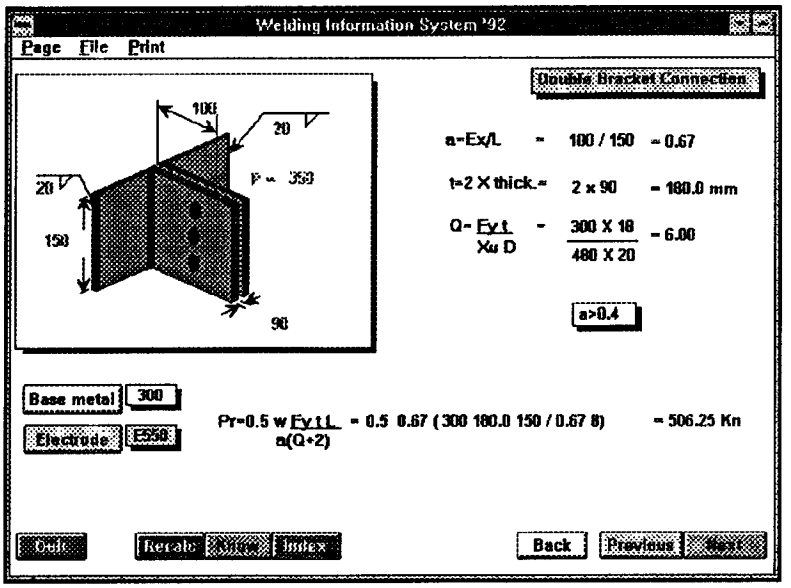

Fig 4.17 Double Angle Connection Template.

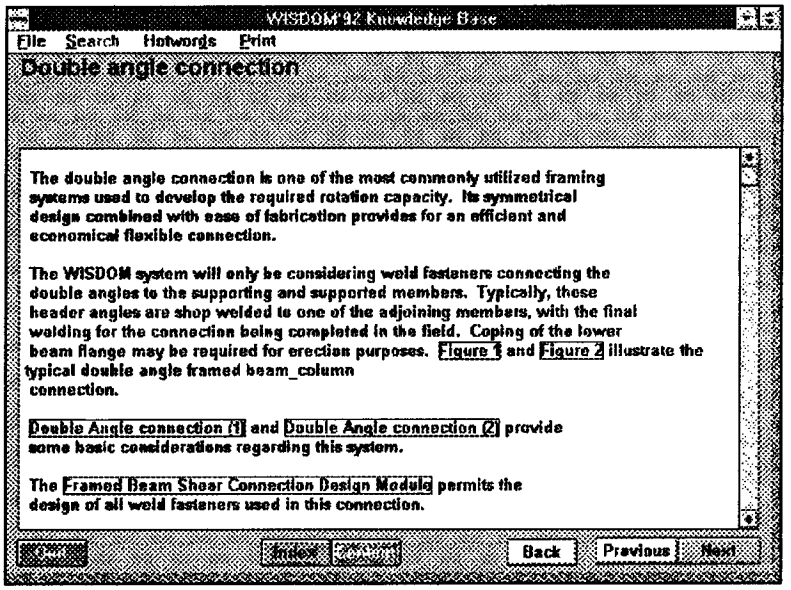

Fig 4.18 Knowledge Base Page.

4.3.4 THE INDEX BUTTON

This button will take the user to the WISDOM main index.

4.3.5 THE BACK BUTTON

The WISDOM system automatically keeps track of the topic the user has visited. Clicking the *back button* will take him to the previous topic he was exploring.

4.3.6 THE PREVIOUS BUTTON AND NEXT BUTTON

These buttons take you to the next or previous topic in the program. The effect is like flipping pages in a book. When the user has reached one of "the covers" and cannot go farther, then the button's label turns white.

4.4 THE KNOWLEDGE BASE

Another important part of the WISDOM system is the large knowledge base on welding basics, welding procedures, connections, etc. This feature is accessible through the use of the *know button* of the main program or the knowledge base button situated in the main index. With the help of a context sensitive link this sub program is activated. The knowledge base will automatically jump to the topic of interest. If not, it will go to the knowledge base index.

4.4.1 TYPICAL KNOWLEDGE BASE PAGE

A typical example would be: a user who is working in the design template environment completing the sizing of a double angle connection (Fig. 4.17). He wants to know more about this type of connection and presses the *know button*. This action will activate the knowledge base; it appears as a screen with three main parts: a title bar, main text field and standard navigation bar (Fig. 4.18). The title bar describes the subject content of the main text field that may or may not have a scroll bar. These text fields can contain "hotwords" (Fig.4.19). The user can navigate through the text and also show related figures or diagrams by clicking on these (Fig. 4.20).

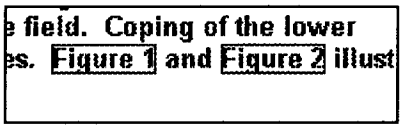

Fig. 4.19 Hotwords.

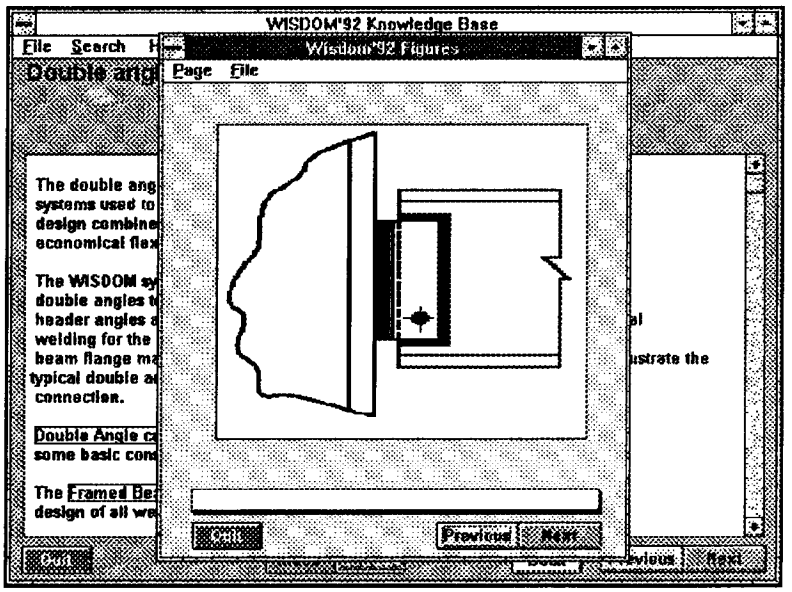

Fig. 4.20 Picture library.

4.4.2 THE KNOWLEDGE BASE CONTENT

The WISDOM system contains a variety of information on welding and welding processes. This knowledge base exists in a hypertext form, which does not have a very rigid structure compared to more traditional texts. However, it contains several subject blocks that interrelate by hypertext links. The subject matters will not all be dicussed in detail This can best be seen by browsing through the hypertext. A short description of the main blocks might be useful, however general welded construction are discussed in the one of the main blocks. The topics contain subject matter on the pros and cons of using welding in construction. Another part describes the more fundamental topics like "steel" and the many types of production methods used to fabricate it: blast furnaces, vacuum furnaces, etc. Also, there are general notes on steel properties: phase diagrams, metallurgy, etc. Another block of topics encompasses welding basics. There is a short historical background of welding processes, and a description of the main weld types: fillet, plug, etc. The next section gives short descriptions of welding defects, problems in welding thick and thin plates. This section also contains topics on design considerations: steel grade choice, electrode choice, detailing practices. Descriptions of popular connections are included with short texts on their advantages and limitation.

A great number of topics are covered in the text. They provide an extensive reference on a wide range of questions that a designer might encounter in his daily practice.

4.4.3 THE KNOWLEDGE BASE INDEX AND TABLE OF CONTENT

The knowledge base index (Fig. 4.21) helps a user find a specific topic. It includes the complete list of titles in the top left corner of the screen. The use of this list is quite simple: search out the topic of interest by using the scroll bar or the alphabetical search (the letter bar at the bottom). Double click on the choice and the display proceeds to that topic of the knowledge base.

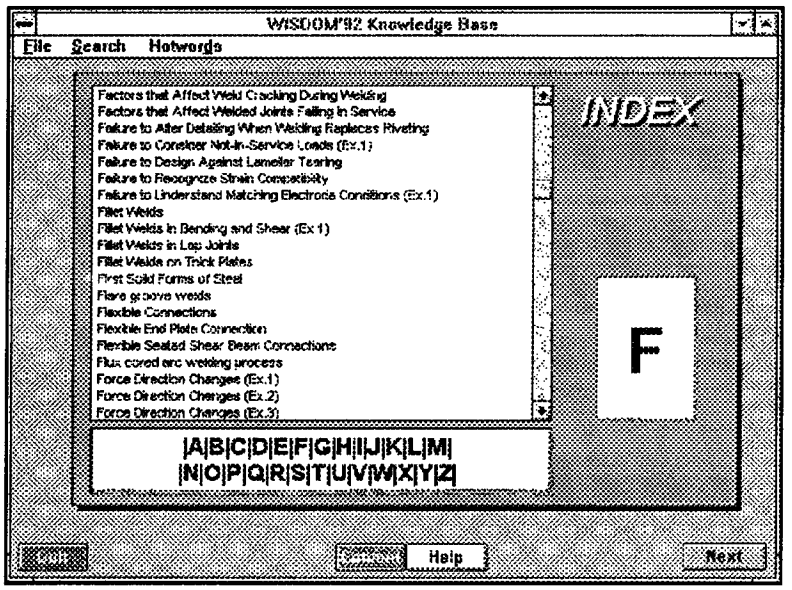

Fig. 4.21 Index of subjets.

WISDOM also furnishes an active table of content (Fig. 4.22). Its use is quite similar to that of the index. The user can browse through all the subject matter of the knowledge base, then go to a topic of interest by clicking on the "hotword" that describes it.

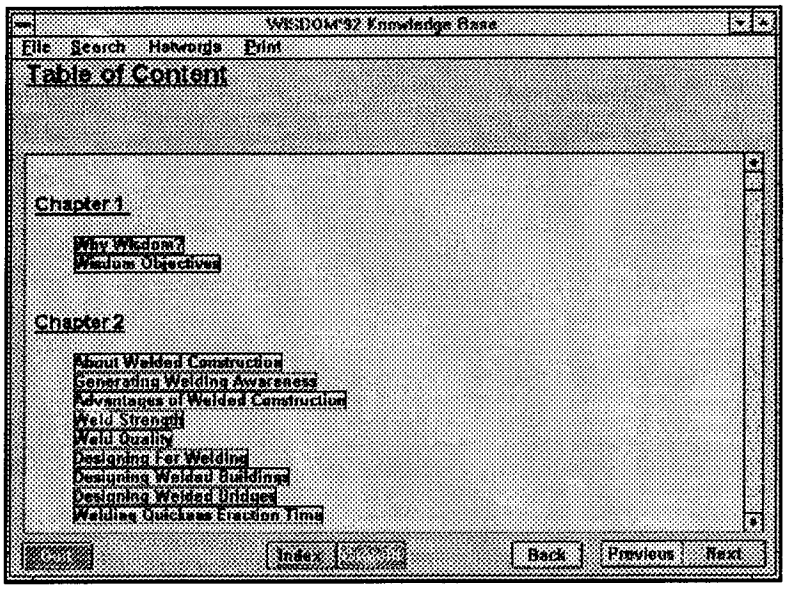

Fig. 4.22 Table of content.

4.5 THE FIGURE LIBRARY

This library (Fig. 4.20) contains over 200 diagrams, graphs and connection details. These pictures are accessed through the hypertext of the knowledge base. When a user clicks on one of the hotwords related to a figure: a small windows containing the picture of interest overlays the existing screens. A navigation buttons is located at the bottom of the page so the user can flip through the diagrams. The user can exit this module by pressing the *QUIT button.*

SUMMARY

The primary objective of the **WISDOM** is to promote welded design through welding awareness. Many young engineers are coming out of their studies with only limited knowledge of welding techniques. This in turn makes some uncomfortable about using even the most basic welding design. It was thought, that by using a program that facilitates calculations and provides ways for a person to quickly learn more about the subject at hand a greater general quality could be achieved in the industry. With the interactive knowledge base the software achieves this objective. For example, a person can look for specific piece of information or read about the whole iron making process.

Another objective, visualization of the problem, was attained by using a Microsoft WINDOWS based programming language. Because of this, the program's framework becomes very graphical, and all of the commands are very simple to use. The complex interactions between the modules are hidden in the background, thus enabling the user to concentrate on the task at hand without worrying about the overhead. This gives him/her more time to understand the different variables that make a good welded design. If a designer wants to learn more about a problem, the information is at his fingertips.

WISDOM was not meant to end at this stage. Because of its open structure, this program can easily be expanded. A number of design aides can be added to the existing framework or other modules can be developed independently and "hooked-up" to the existing program. Also, with the advancements in technology, one can foresee enhancements in the existing structure that drives the software. CD-ROM with its mass storage capabilities will probably be used to store short video clips and sound effects to render subjects even more effectively. A student or professional will, for example, be able

to see on screen different welding machinery at work, short instructional clips, etc. The more technology advances the more one needs tools to deal with the information that is generated by it.

This pilot project was an attempt at using non-traditional information organization on a classic textbook subject. Engineers are asked to know more and more, and learn it in a shorter period of time. They have to start at looking at different techniques of learning to help the traditional textbook approach, because they will probably have to deal with information overload sooner than later. The WISDOM project attempted to address this problem head on. Through new data structuring techniques and the use of the WINDOWS environment, the result of this work was a very graphical program with a large and easy to use knowledge base. The main conclusion of this work is that: more work has to be done in the field of knowledge organization, because the capacity of the system is growing faster than the capabilities of humans to systematize it.

Bibliography

1. SCHADER, M., GAUL, W., "Knowledge, Data and Computer-Assisted Decisions", NATO Advance Research Workshop on Data, Expert Judgment and Decision, Spinger-Verlag, Berlin New-York, viii, 1990.

2. MUMPOWER et Al., "Expert Judgement and Expert Systems", NATO Advance Research Workshop on Data, Expert Judgment and Decision, Springer-Verlag, viii, Berlin New-York, 1987.

3. WATERWORTH, J., "Multimedia: Technology and Applications", Ellis Horwood, New York, 1991.

4. COTTON, B., OLIVER, R., "Understanding Hypermedia: From Multimedia to Virtual Reality", Phaidon Press, London, 1993.

5. NEWELL, A., SIMON, H.A., "Human Problem Solving [by] Allen Newell [and] Herbert A. Simon", Prentice Hall Inc., Englewood Cliffs, N.J., 1972.

6. Canadian Institute of Steel Constrution, "Hanbook of Steel Contruction", Willowdale, Ontario, october, 1991.

7. Canadian Standards Association, "Limit States Design of Steel Structures", CAN/CSA-S16.1-M89, Rexdale, Ontario, 1989.

8. Canadian Standards Association, "Welded Steel Construction (Metal Arc Welding)", CSA W59-M1989, Rexdale, Ontario, 1989.

9. PICARD A., BEAULIEU D., "Calcul aux États Limites des Charpentes D'Acier", Institut Canadien de la Construction en Acier, Willowdale, Ontario, 1991.

10. Butler, Lorne J.., Pal, Shudendu and Kulak, Geoffrey L., "Eccentrically Loaded Welded Connections", Journal of the Structural Division, Proceedings of ASCE, May, 1972

11. Bush, Vannevar," As We May Think",Atlantic Monthly, January, 1945

Appendices A

WISDOM Scripts

Wisdom.tbk book script

```
to handle keyDown key
  if key is keyF3
    send author
    break
  end
  forward
end

-- change default setting of text when entering book
to handle enterBook
        restore system
        go page 1
        set sysFontFace to arial
   set sysFontSize to 10
   set sysFontStyle to bold
   set sysCursor to 1
   set sysDecimal to "."
   set sysTabSpacing to 1584
        set syschangesdb to false
        hide scrollbar
-- let's change the menu
   restore menuBar at both
   remove menu "file" at reader
   remove menu "edit" at reader
   remove menu "text" at reader
   remove menu "help" at reader
   add menu "&File"  at reader
       add menuItem "&Compact Book" to menu "&File" at reader
   add menuItem "&Quit"  to menu "&File" at reader
   add menu "&Print"  at reader
   add menuItem "&Print Page" alias "printPage" to menu "Print" at reader
   add menuItem "Print &Results" alias "printresults" to menu "print" at reader
   deactivate menuItem "printPage" at reader
   deactivate menuItem "printresults" at reader
   remove menuitem "NewPage" at reader
        send reader

        linkDll "WISDOM.dll"
           long boxAllC (double,double,double)
           long cweldcc (double,double,double)
           long cweldoc (double,double,double)
           long lweldcc (double,double,double)
           long lweldoc (double,double,double)
           long parhorc (double,double,double)
           long parverc (double,double,double)
           long outplaneverc (double,double,double,double,double)

        end linkdll
```

```
-- firstTime is a book property, if it set to true, the first time a
-- user opens Wisdom he will be prompted to see if he wants to take
-- the quick tour. Before sending out wisdom, the command window should
-- be used to set the propertie to true.
--                    set firstTime of this book to true


        get firstTime of this book
        if it is true
    request "Do you want to take a quick tour of WISDOM'92 first ?" \
              with "OK" or "Cancel"
    if it is "OK"
       set the firstTime of this book to false
       send startExe "wistour"
       send save
       send exit
       break to system
    end
   end
end


to handle printPage

   get name of this background
   if it is "normal"
     hide group "buttons" of this background
     set printborder to false
     start spooler
       print 1
     end spooler
     show group "buttons" of this background
   end if
end

to handle printResults

  set printerFields to "viewr"
  set printerStyle to groups
  set printerFieldNames to false
  set printerFieldWidths to 9360
  set printerRightMargin to 1440

  start spooler
    print 1
  end spooler

end


to handle doInfo ContextInfo

-- start workhelp
   send startExe "workhelp"
-- make workhelp find the topic
```

```
    executeRemote " send doinfo "&&contextInfo \
        application toolbook topic "workhelp.tbk"
    set syssuspend to true
    if item 1 of syserror is not "ok" then
       request item 1 of syserror
       break
    end if
    set syscursor to default
end

to handle doFigures
    send startExe "wpicture"
    set syssuspend to true
    if item 1 of syserror is not "ok" then
       request item 1 of syserror
       break
    end if
    set syscursor to default
end

to handle doProcedures
    send startExe "wform"
    set syssuspend to true
    if item 1 of syserror is not "ok" then
       request item 1 of syserror
       break
    end if
    set syscursor to default
end

to handle startExe ExeName
    set syscursor to 4
    -- see if workhelp is running
    getRemote "sysWindowHandle" application "toolbook" topic \
       ExeName&".TBK"
    set hWnd to it
    -- if it is not then run it
    if hWnd is null
      --find ExeName.tbk
      clear sysError
      set syssuspend to false
      get  ExeName&".tbk"
      get book it
      set syssuspend to true
      if syserror is not null
         set syscursor to 1
         request  strNoBook(ExeName)
         break to system
      end
      --try running toolbook and the app
      set syssuspend to false
      clear syserror
      if sysruntime is true
         set exe to "tbook.exe"
         run "tbook.exe"&& ExeName&".tbk"
      else
```

```
            set exe to "toolbook.exe"
            run "toolbook.exe"&& ExeName&".tbk"
        end
        set syssuspend to true
        if syserror is not null
          set syscursor to 1
          request strExeNotInPath(exe)
          break to system
        end
      else
       -- maximize the window
            linkDLL "user"
                  INT  ShowWindow(WORD,INT)
            end linkDLL
        get ShowWindow(hWnd,3)
        unlinkDLL "user"
      end if
      set syssuspend to false
      clear syserror
      set syscursor to default

end


to handle author
   set sysChangesDB to true
   show menubar
   send sizeToPage
   forward
end


to handle reader
   set syschangesdb to false
   send sizeToPage
   forward
end


--change default setting to system default when leaving
to handle leaveBook
  set  syslockscreen to true
  unlinkDLL "WISDOM.DLL"
  restore system
end

to handle quit
--   send save
  send exit
end

to handle GoToPage dest
   go to page dest
end

--   Book strings
```

```
to get strExeNotInPath exe
   return "The directory for" && exe && "is not in your current path.  Change your path to include" && exe
& "'s directory."
end

to get strNoBook ExeName
   return "Can't find the book. Please copy"&& ExeName &".tbk into this directory."
end

to get strNumericValue
   return "Please enter a numeric value."
end

to get strCancel
   return "Cancel"
end

to get strOK
   return "OK"
end

to get strNotInPath
   return "Can't find Workhelp"
end

to get strGood  r,c
    return "Welded connection's resistance is enough ! "&c&"kN > "&R&"kN"
end

to get strNoGood r,c
    return "Welded connection's resistance is not enough ! "&R&"kN > "&c&"kN"
end


--squeeze the hell out of a modified book
--don't forget to change to the correct name of this book!
to handle compactBook
  get name of this book
  if it is null
     request "Can't compact a untitled book."
     break to system
  end if
  clear sysError
  save as "X.TBK",true
  if sysError is not null
     request "Save as X.TBK failed:"&&sysError
     break to system
  end if
  save as it ,true
  if sysError is not null
     request "Save as original name failed:"&&sysError
     break to system
  end if
end compactBook
```

Script of Page double Angle Connection

```
to handle leavefield

    if focus is not null
      get text of target          --localize the text of the field
      set syssuspend to false     --turn this off so no error messages
      set syserror to null        --initialize
      increment it by 1           --try a numeric operation on the text
      set syssuspend to true      --don't forget to turn this on again
      if syserror is not null     --test for an error
         request strNumericValue() with strOK() or strCancel()
         if it is strCancel()
            set syssuspendmessages to true
            set focus to null
            set syssuspendmessages to false
            break
         end
         break to system
      end
    end if

    put typeMet of button "metal" into fy
    put typeElec of button "electrode" into xu
    set qro to ((0.31737*xu)/1000)
    format number qro as "0" & sysdecimal & "000" from null
    local lg,ex,ep,gr,a,t,q,qualite1,qualite2,fr
    put the text of field "longeur" into lg
    put the text of field "excen"   into ex
    put the text of field "epais"   into ep
    put the text of field "grosseur" into gr
    put the text of field "force" into fr
  -- calculation of basic values and formating
    set a to (ex/lg)
    format number a as "0" & sysDecimal & "00" from null
    set t to (2*ep)
    format number t as "0" & sysDecimal & "0" from null
    set q to round((fy*t)/(xu*gr))
    format number q as "0" & sysDecimal & "00" from null
  -- put the entered values into the visible fields
    put the (text of field "excen"&" / "& text of field "longeur") into \
    text of field "champs1"
    put  ("2 x "& text of field "epais") into the text of field "champs2"
    put  (fy&&"X"&&t) into the text of field "champs3"
    put  (xu&&"X"&&gr) into the text of field "champs4"
    put   ("="&&a) into the text of field "champs5"
    put   ("="&&t&&"mm") into the text of field "champs6"
    put   ("="&&q) into the text of field "champs7"
  -- control to see what equation template needs to be shown
    if a >= 0.4 then
      set qualite1 to true
      set qualite2 to false
    else
       set qualite1 to false
       set qualite2 to true
    end if
```

```
    step i from 1 to 4
      set it to "equasup"&i
      put qualite1 into visible of field it
    end step
    step i from 5 to 14
       set it to "equasup"&i
      put qualite2 into visible of field it
    end step
    if a >= 0.4 then
      put ("0.5 "&" 0.67 ("&&fy&&t&&lg&" / "&a&&(q+2)&")") into the text of field equasup2
      set re1 to ((0.5*0.67*((fy*t*lg)/(a*(q+2))))/1000)
      format number re1 as "0" & sysDecimal & "00" from null
      put ("="&&re1&&"Kn") into text of field equasup3
      put re1 into it
    else
      set re2 to ((0.5*0.67*((fy*t*lg)/(0.4*(q+2))))/1000)
      format number re2 as "0" & sysDecimal & "00" from null
      set re3 to (2*qro*gr*lg)
      format number re3 as "0" & sysDecimal & "00" from null
      set re4 to (re3+((a^2)*(re2-re3)/0.16))
      format number re4 as "0" & sysDecimal & "00" from null
      put ("0.5X"&" 0.67 ("&&fy&"X"&t&"X"&lg&" / "&"0.4X"&&(q+2)&")") into the text of field "equasup8"
      put ("2 X"&qro&" X "&gr&" X "&lg) into the text of field "equasup9"
      put (re2&" + ("&a&"²("&re2&" -"&re3&")/0.16") into the text of field "equasup10"
      put ("="&&re2&&"kN") into the text of field "equasup11"
      put ("="&&re3&&"kN") into the text of field "equasup12"
      put ("="&&re4&&"kN") into the text of field "equasup13"
      put re4 into it
    end if
    if (it > fr ) then
       hide paintObject "connec2"
       show paintobject "connec1"
    else
       hide  paintobject "connec1"
       show  paintobject "connec2"
    end if

end
to get strNumericValue
    return "Please enter a numeric value."
end

to get strCancel
    return "Cancel"
end

to get strOK
    return "OK"
end
```

Script for Button Base Metal

```
to handle buttonUp

    send startExe "wbasemet"
```

```
    put the name of this page into it
    set the prpage of button "ok" of page "tableau" of book "wbasemet.tbk" to it


end

to handle writeField
  get my typeMet
  put it into the text of field "typeMet" of this page
end
```

Script for Button Electrode

```
to handle buttonUp

  send startExe "welectro"
  put the name of this page into it
  set the prpage of button "ok" of page "tableau" of book "welectro.tbk" to it


end


to handle writeField
  get my typeElec
  put "E"&it into the text of field "typeElec" of this page
end
```

Script for Tension Member

```
to handle leavefield


   if focus is not null
     get text of target          --localize the text of the field
     set syssuspend to false     --turn this off so no error messages
     set syserror to null        --initialize
     increment it by 1           --try a numeric operation on the text
     set syssuspend to true      --don't forget to turn this on again
     if syserror is not null     --test for an error
        request strNumericValue() with strOK() or strCancel()
        if it is strCancel()
           set syssuspendmessages to true
           set focus to null
           set syssuspendmessages to false
           break
        end
        break to system
     end
   end if

  local re1,xu,y1,y2,l1,l2,gro,qro
```

```
    put typeElec of button "electrode" into xu
    put typemet of button "metal" into fy
    set qro to ((0.3535*xu)/1000)
    format number qro as "0" & sysdecimal & "000" from null
    put ("0.3535 "&xu&&"=") into text of field "champs6"
    put (qro&&"kN/mm^2") into the text of field "champs7"
    put the text of field "y1" into y1
    put the text of field "y2" into y2
    put the text of field "long1" into l1
    put the text of field "sou1" into gro
    put the text of field "force" into fr
  -- calculation of basic values and formating
    set l2 to ((y1/y2)*l1)
    format number l2 as "0" & sysDecimal & "0" from null
    put l2 into the text of field "long2"
  -- put the entered values into the visible fields
    put (gro&"  "&qro&&"("&&l1&&"+"&&l2&&") =") into the text of field "champs3"
    set re1 to (gro*qro*(l1+l2))
    format number re1 as "0" & sysdecimal & "00" from null
    put (re1&&"kN") into the text of field "champs4"
  -- base metal resistance
    set re2 to ((0.67*0.9*gro*(l1+l2)*fy)/1000)
    format number re2 as "0" & sysdecimal & "00" from null
    put ("0.67 0.9 "&gro&" ( "&l1&&"+"&&l2&&") "&&fy) into the text of field "champs5"
    put (re2&&"kN") into the text of field "re2"

    if ((re1 < fr) or (re2 < fr)) then
      show paintobject "connec2"
      hide paintobject "connec1"
    else
      show paintobject "connec1"
      hide paintobject "connec2"
    end if
  end
```

Script Two Vetical Welds Plastic

```
to handle leaveField
    local e,w,l,d


    if focus is not null
      get text of target          --localize the text of the field
      set syssuspend to false     --turn this off so no error messages
      set syserror to null        --initialize
      increment it by 1           --try a numeric operation on the text
      set syssuspend to true      --don't forget to turn this on again
      if syserror is not null     --test for an error
         request strNumericValue() with strOK() or strCancel()
         if it is strCancel()
            set syssuspendmessages to true
            set focus to null
            set syssuspendmessages to false
            break
```

```
        end
        break to system
      end
    end if
    forward leavefield
    put the text of field "e" into e
    put the text of field "w" into w
    put the text of field "l" into l
    put the text of field "d" into d
--  calculation of C by DLL
    set C to parverc(l,w,e)
    set c to (c/1000)
    set pr to (c*d*l)
    format number C as "0" & sysDecimal & "000" from null
    format number pr as "0" & sysDecimal & "0" from null
    put (c&&"X"&&d&&"X"&&l) into the text of field "Pr1"
    put ("="&&pr&&"kN") into the text of field "Pr"
    put (pr) into the text of field "Cr1"
    put (d&&"X"&&l) into the text of field "Cr2"
    put ("="&&c) into the text of field "Cr"
    put (pr) into the text of field "Dr1"
    put (c&&"X"&&l) into the text of field "Dr2"
    put ("="&&d&&"mm") into the text of field "Dr"
    put (pr) into the text of field "Lr1"
    put (c&&"X"&&d) into the text of field "Lr2"
    put ("="&&l&&"mm") into the text of field "Lr"

-- result field
    put the text of field "p" into p
    set the text of field "PRV" to (pr&&"kN")
    set the text of field "PV" to (p&&"kN")
    if (pr>=p) then
      set the text of field "signe" to ">"
      set fieldColor to "240,50,100"
    else
      set the text of field "signe" to "<"
      set fieldColor to "0,50,100"
    end
    set list to objects of group "view"
    while list is not null
      pop list
      set strokeColor of it to fieldColor
    end


end
```

Script For Two Vertical Welds Plastic

```
to handle leavefield


   if focus is not null
     get text of target          --localize the text of the field
```

```
  set syssuspend to false    --turn this off so no error messages
  set syserror to null       --initialize
  increment it by 1          --try a numeric operation on the text
  set syssuspend to true     --don't forget to turn this on again
  if syserror is not null    --test for an error
     request strNumericValue() with strOK() or strCancel()
     if it is strCancel()
        set syssuspendmessages to true
        set focus to null
        set syssuspendmessages to false
        break
     end
     break to system
  end
 end if
 forward leavefield
 put the text of field "vfx" into vfx
 put the text of field "vfy" into vfy
 put the text of field "lx" into lx
 set lx to (-lx)
 put the text of field "ly" into ly
 set ly to (-ly)
 put the text of field "l" into l
 put the text of field "b" into b
 put the text of field "d" into d
 set xu to typeElec of button "electrode" of this page

 set qr to ((0.67*0.67*0.707*xu)/1000)
 set vr to (qr*d)
 format number vr as "0" & sysdecimal & "000" from null
 get ("0.67X0.67X0.707X"&xu&"X"&d)
 set text of field "VVR" to it
 set text of field "VR" to ("="&vr&&"kN/mm")
 set ix to ((l^3)/6)
 format number ix as "0" from null
 get (l&"^3/6")
 set text of field "vix" to it
 set text of field "ix" to ("="&ix&&"mm^3")
 set iy to ((l/6)+((l*(b^2))/2))
 format number iy as "0" from null
 get (l&"/6 + "&l&" X "&b&"^2/2")
 set text of field "viy" to it
 set text of field "iy" to ("="&iy&&"mm^3")
 set ip to (ix+iy)
 format number ip as "0"  from null
 get (ix&&"+"&&iy)
 set the text of field "vip" to it
 set the text of field "ip" to ("="&ip&&"mm^3")
 set vx to abs(vfx/(2*l))
 format number vx as "0" & sysdecimal&"000" from null
 get ("¦"&vfx&"/(2 X "&l&")¦")
 put it into the text of field "vvx"
 set the text of field "vx" to ("="&vx&&"kN/mm")
 set vy to abs(vfy/(2*l))
 format number vy as "0" & sysdecimal&"000" from null
 get ("¦"&vfy&"/(2 X "&l&")¦")
```

```
    put it into the text of field "vvy"
    set the text of field "vy" to ("="&vy&&"kN/mm")
    set vxb to abs(((vfx*ly+vfy*lx)*(l/2))/ip)
    format number vxb as "0"&sysdecimal&"000" from null
    get ("¦("&vfx&"X"&ly&"+"&vfy&"X"&lx&")X("&l&"/2)/"&ip&")¦")
    put it into the text of field "vvxb"
    set the text of field "vxb" to ("="&vxb&&"kN/mm")
    set vyb to abs(((vfx*ly+vfy*lx)*(b/2))/ip)
    format number vyb as "0"&sysdecimal&"000" from null
    get ("¦("&vfx&"X"&ly&"+"&vfy&"X"&lx&")X("&b&"/2)/"&ip&")¦")
    put it into the text of field "vvyb"
    set the text of field "vyb" to ("="&vyb&&"kN/mm")
    set v to sqrt((vx+vxb)^2+(vy+vyb)^2)
    format number v as "0"&sysdecimal&"000" from null
    get (("&vx&"+"&vxb&")^2+("&vy&"+"&vyb&")^2)^½)")
    put it into the text of field "vv"
    set the text of field "v" to ("="&v&&"kN/mm")

    set chob to objects of group "results"
    if (v>vr)
      while chob is not null
        pop chob
        set strokecolor of it to red
      end while
    else
      while chob is not null
        pop chob
        set strokecolor of it to blue
      end while
    end if

end
```

Script for Two Horizontal Welds Plastic

```
to handle leavefield


   if focus is not null
     get text of target          --localize the text of the field
     set syssuspend to false     --turn this off so no error messages
     set syserror to null        --initialize
     increment it by 1           --try a numeric operation on the text
     set syssuspend to true      --don't forget to turn this on again
     if syserror is not null     --test for an error
        request strNumericValue() with strOK() or strCancel()
        if it is strCancel()
           set syssuspendmessages to true
           set focus to null
           set syssuspendmessages to false
           break
        end
        break to system
     end
```

```
end if
forward leavefield
put the text of field "vfx" into vfx
put the text of field "vfy" into vfy
put the text of field "lx" into lx
set lx to (-lx)
put the text of field "ly" into ly
set ly to (-ly)
put the text of field "l" into l
put the text of field "b" into b
put the text of field "d" into d
set xu to typeElec of button "electrode" of this page

set qr to ((0.67*0.67*0.707*xu)/1000)
set vr to (qr*d)
format number vr as "0" & sysdecimal & "000" from null
get ("0.67X0.67X0.707X"&xu&"X"&d)
set text of field "VVR" to it
set text of field "VR" to ("="&vr&&"kN/mm")
set ix to ((l/6)+((l*(b^2))/2))
format number ix as "0" from null
get (l&"/6 + "&l&" X "&b&"^2/2")
set text of field "vix" to it
set text of field "ix" to ("="&ix&&"mm^3")
set iy to ((l^3)/6)
format number iy as "0" from null
get (l&"^3/6")
set text of field "viy" to it
set text of field "iy" to ("="&iy&&"mm^3")
set ip to (ix+iy)
format number ip as "0"  from null
get (ix&&"+"&&iy)
set the text of field "vip" to it
set the text of field "ip" to ("="&ip&&"mm^3")
set vx to abs(vfx/(2*l))
format number vx as "0" & sysdecimal&"000" from null
get ("¦"&vfx&"/(2 X "&l&")¦")
put it into the text of field "vvx"
set the text of field "vx" to ("="&vx&&"kN/mm")
set vy to abs(vfy/(2*l))
format number vy as "0" & sysdecimal&"000" from null
get ("¦"&vfy&"/(2 X "&l&")¦")
put it into the text of field "vvy"
set the text of field "vy" to ("="&vy&&"kN/mm")
set vxb to abs(((vfx*ly+vfy*lx)*(l/2))/ip)
format number vxb as "0"&sysdecimal&"000" from null
get ("¦("&vfx&"X"&ly&"+"&vfy&"X"&lx&")X("&l&"/2)/"&ip&")¦")
put it into the text of field "vvxb"
set the text of field "vxb" to ("="&vxb&&"kN/mm")
set vyb to abs(((vfx*ly+vfy*lx)*(b/2))/ip)
format number vyb as "0"&sysdecimal&"000" from null
get ("¦("&vfx&"X"&ly&"+"&vfy&"X"&lx&")X("&b&"/2)/"&ip&")¦")
put it into the text of field "vvyb"
set the text of field "vyb" to ("="&vyb&&"kN/mm")
set v to sqrt((vx+vxb)^2+(vy+vyb)^2)
format number v as "0"&sysdecimal&"000" from null
```

```
  get ("(("&vx&"+"&vxb&")^2+("&vy&"+"&vyb&")^2)^½)")
  put it into the text of field "vv"
  set the text of field "v" to ("="&v&&"kN/mm")


    set chob to objects of group "results"
  if (v>vr)
    while chob is not null
      pop chob
      set strokecolor of it to red
    end while
  else
    while chob is not null
      pop chob
      set strokecolor of it to blue
    end while
  end if


end
```

Script For Box Weld Elastic

```
to handle leavefield


  if focus is not null
    get text of target          --localize the text of the field
    set syssuspend to false     --turn this off so no error messages
    set syserror to null        --initialize
    increment it by 1           --try a numeric operation on the text
    set syssuspend to true      --don't forget to turn this on again
    if syserror is not null     --test for an error
       request strNumericValue() with strOK() or strCancel()
       if it is strCancel()
          set syssuspendmessages to true
          set focus to null
          set syssuspendmessages to false
          break
       end
       break to system
    end
  end if
  forward leavefield
  put the text of field "vfx" into vfx
  put the text of field "vfy" into vfy
  put the text of field "lx" into lx
  set lx to (-lx)
  put the text of field "ly" into ly
  set ly to (-ly)
  put the text of field "l" into l
  put the text of field "b" into b
  put the text of field "d" into d
  set xu to typeElec of button "electrode" of this page
```

```
 set qr to ((0.67*0.67*0.707*xu)/1000)
 set vr to (qr*d)
 format number vr as "0" & sysdecimal & "000" from null
 get ("0.67X0.67X0.707X"&xu&"X"&d)
 set text of field "VVR" to it
 set text of field "VR" to ("="&vr&&"kN/mm")
 set ix to ((l^3/6)+(b/6)+(b*l^2/2))
 format number ix as "0" from null
 get (l&"^3/6 + "&b&"/6+"&b&"X"&l&"^2/2")
 set text of field "vix" to it
 set text of field "ix" to ("="&ix&&"mm^3")
 set iy to ((l/6)+(l*b^2/2)+(b^3/6))
 format number iy as "0" from null
 get (l&"/6+"&l&"X"&b&"^2/2+"&b&"^3/6")
 set text of field "viy" to it
 set text of field "iy" to ("="&iy&&"mm^3")
 set ip to (ix+iy)
 format number ip as "0"  from null
 get (ix&&"+"&&iy)
 set the text of field "vip" to it
 set the text of field "ip" to ("="&ip&&"mm^3")
 set vx to abs(vfx/(2*(l+b)))
 format number vx as "0" & sysdecimal&"000" from null
 get ("¦"&vfx&"/(2 X("&l&"+"&b&"))¦")
 put it into the text of field "vvx"
 set the text of field "vx" to ("="&vx&&"kN/mm")
 set vy to abs(vfy/(2*(l+b)))
 format number vy as "0" & sysdecimal&"000" from null
 get ("¦"&vfy&"/(2 X("&l&"+"&b&"))¦")
 put it into the text of field "vvy"
 set the text of field "vy" to ("="&vy&&"kN/mm")
 set vxb to abs(((vfx*ly+vfy*lx)*(l/2))/ip)
 format number vxb as "0"&sysdecimal&"000" from null
 get ("¦("&vfx&"X"&ly&"+"&vfy&"X"&lx&")X("&l&"/2)/"&ip&")¦")
 put it into the text of field "vvxb"
 set the text of field "vxb" to ("="&vxb&&"kN/mm")
 set vyb to abs(((vfx*ly+vfy*lx)*(b/2))/ip)
 format number vyb as "0"&sysdecimal&"000" from null
 get ("¦("&vfx&"X"&ly&"+"&vfy&"X"&lx&")X("&b&"/2)/"&ip&")¦")
 put it into the text of field "vvyb"
 set the text of field "vyb" to ("="&vyb&&"kN/mm")
 set v to sqrt((vx+vxb)^2+(vy+vyb)^2)
 format number v as "0"&sysdecimal&"000" from null
 get (("&vx&"+"&vxb&")^2+("&vy&"+"&vyb&")^2)^½)")
 put it into the text of field "vv"
 set the text of field "v" to ("="&v&&"kN/mm")

    set chob to objects of group "results"
 if (v>vr)
   while chob is not null
     pop chob
     set strokecolor of it to red
   end while
 else
   while chob is not null
```

```
      pop chob
      set strokecolor of it to blue
    end while
  end if

end
```

Script for C Weld Elastic

```
to handle leavefield


  if focus is not null
    get text of target          --localize the text of the field
    set syssuspend to false     --turn this off so no error messages
    set syserror to null        --initialize
    increment it by 1           --try a numeric operation on the text
    set syssuspend to true      --don't forget to turn this on again
    if syserror is not null     --test for an error
      request strNumericValue() with strOK() or strCancel()
      if it is strCancel()
        set syssuspendmessages to true
        set focus to null
        set syssuspendmessages to false
        break
      end
      break to system
    end
  end if
  forward leavefield
  put the text of field "vfx" into vfx
  put the text of field "vfy" into vfy
  put the text of field "lx" into lx
  set lx to (-lx)
  put the text of field "ly" into ly
  set ly to (-ly)
  put the text of field "l" into l
  put the text of field "b" into b
  put the text of field "d" into d
  set xu to typeElec of button "electrode" of this page

  set x to ((b^2)/(l+2*b))
  format number x as "0"&sysdecimal&"0" from null
  set text of field "x" to x
  set qr to ((0.67*0.67*0.707*xu)/1000)
  set vr to (qr*d)
  format number vr as "0" & sysdecimal & "000" from null
  get ("0.67X0.67X0.707X"&xu&"X"&d)
  set text of field "VVR" to it
  set text of field "VR" to ("="&vr&&"kN/mm")
  set ix to ((l^3/12)+(b/6)+(b*l^2/2))
  format number ix as "0" from null
  get (l&"^3/12 + "&b&"/6+"&b&"X"&l&"^2/2")
  set text of field "vix" to it
```

```
 set text of field "ix" to ("="&ix&&"mm^3")
 set iy to ((l/12)+(l*x^2)+(2*((b^3/12)+(b*((b/2)-x)^2))))
 format number iy as "0" from null
 get (l&"/12+"&l&"X"&x&"^2+2X("&b&"^3/12+"&b&"X("&b&"/2-"&x&")^2)")
 set text of field "viy" to it
 set text of field "iy" to ("="&iy&&"mm^3")
 set ip to (ix+iy)
 format number ip as "0"  from null
 get (ix&&"+"&&iy)
 set the text of field "vip" to it
 set the text of field "ip" to ("="&ip&&"mm^3")
 set mf to (abs(vfx)*ly+abs(vfy)*lx)
 format number mf as "0" from null
 get ("¦"&vfx&"¦X"&ly&"+"&"¦"&vfy&"¦X"&lx)
 set the text of field "vmf" to it
 set the text of field "mf" to ("="&mf&"kN-mm")
 set vx to abs(vfx/(l+2*b))
 format number vx as "0" & sysdecimal&"000" from null
 get ("¦"&vfx&"/("&l&"+2X"&b&")¦")
 put it into the text of field "vvx"
 set the text of field "vx" to ("="&vx&&"kN/mm")
 set vy to abs(vfy/(l+2*b))
 format number vy as "0" & sysdecimal&"000" from null
 get ("¦"&vfy&"/("&l&"+2X"&b&")¦")
 put it into the text of field "vvy"
 set the text of field "vy" to ("="&vy&&"kN/mm")
 set vxb to (mf*(l/2)/ip)
 format number vxb as "0"&sysdecimal&"000" from null
 get (mf&"X("&l&"/2)/"&ip)
 put it into the text of field "vvxb"
 set the text of field "vxb" to ("="&vxb&&"kN/mm")
 set vyb1 to (mf*(b-x)/ip)
 format number vyb1 as "0"&sysdecimal&"000" from null
 get (mf&"X("&b&"-"&x&")/"&ip)
 put it into the text of field "vvyb1"
 set the text of field "vyb1" to ("="&vyb1&&"kN/mm")
 set vyb2 to (mf*x/ip)
 format number vyb2 as "0"&sysdecimal&"000" from null
 get (mf&"X"&x&"/"&ip)
 put it into the text of field "vvyb2"
 set the text of field "vyb2" to ("="&vyb2&&"kN/mm")

 push sqrt((vx+vxb)^2+(vy-vyb1)^2) onto va
 push sqrt((vx-vxb)^2+(vy-vyb1)^2) onto va
 push sqrt((vx-vxb)^2+(vy+vyb2)^2) onto va
 push sqrt((vx+vxb)^2+(vy+vyb2)^2) onto va
 set big to 0
 step i from 4 to 1 by -1
   pop va
   if (it > big) then
      set big to it
      set index to i
   end if
 end step
 conditions
   when (index = 1)
```

```
        get ("(("&vx&"+"&vxb&")^2+("&vy&"-"&vyb1&")^2)^½)")
      when (index=2)
        get ("(("&vx&"-"&vxb&")^2+("&vy&"-"&vyb1&")^2)^½)")
      when (index=3)
        get ("(("&vx&"-"&vxb&")^2+("&vy&"+"&vyb2&")^2)^½)")
      when (index=4)
        get ("(("&vx&"+"&vxb&")^2+("&vy&"+"&vyb2&")^2)^½)")
    end conditions
    set v to big
    format number v as "0"&sysdecimal&"000" from null
    put it into the text of field "vv"
    set the text of field "v" to ("="&v&&"kN/mm")

      set chob to objects of group "results"
    if (v>vr)
      while chob is not null
        pop chob
        set strokecolor of it to red
      end while
    else
      while chob is not null
        pop chob
        set strokecolor of it to blue
      end while
    end if


end
```

Script For Angle.tbk

```
to handle buttonUp
    get book "wisdom.tbk"
    set a to it
    get my angletype
    set the angletype of button "angle" of page "lweld" of a to it
    get  "send showangle to page "&quote&"lweld"&quote&&""
    executeRemote   it  application "toolbook" topic "wisdom.tbk"
    set sysCursor to 4
    send save
    send exit
end



to handle enterBook
    set sysLockScreen to true
    send reader
    hide scrollbar
    send sizeToPage
    set sysCursor to 1
    set sysfontface to arial
    set sysfontstyle to bold
    set sysLockScreen to false
```

```
end


to handle reader
  hide menuBar
  send sizetopage
  forward
end

to handle author
  show menuBar
  send sizetopage
  forward
end

to handle exit
  send save
  forward
end


to handle buttonUp
   set butname to  uniquename of target
   set angletype of button "ok" to name of target
   set checked of  butname to true
   set a to my objects
   while a is not null
     pop a
     if (it  is not  butname) then
        set checked of it to false
     end if
   end while

end
```

Script for L Weld

```
to handle enterpage
   system angleGroup
   set angleGroup to angleType of button "angle" of this page
end



to handle showAngle
   system angleGroup
   set syslockscreen to true
   get angleType of button "angle" of this page
   hide group anglegroup
   show group it
   set anglegroup to it
   set syslockscreen to false

end
```

```
to handle leavefield

  --check for a numeric value

  if focus is not null
    get text of target          --localize the text of the field
    set syssuspend to false     --turn this off so no error messages
    set syserror to null        --initialize
    increment it by 1           --try a numeric operation on the text
    set syssuspend to true      --don't forget to turn this on again
    if syserror is not null     --test for an error
       request strNumericValue() with strOK() or strCancel()
       if it is strCancel()
          set syssuspendmessages to true
          set focus to null
          set syssuspendmessages to false
          break
       end
       break to system
    end
  end if
  forward leavefield
  set angleType to angleType of button "angle" of this page
  put the text of field "vfx" of group angletype into vfx
  put the text of field "vfy" of group angletype into vfy
  put the text of field "lx" of group angletype into lx
  set lx to (-lx)
  put the text of field "ly" of group angletype into ly
  set ly to (-ly)
  put the text of field "l" of group angletype into l
  put the text of field "b" of group angletype into b
  put the text of field "d" of group angletype into d
  set xu to typeElec of button "electrode" of this page

  set x to ((b^2)/((l+b))/2)
  format number x as "0"&sysdecimal&"0" from null
  set text of field "x" of group angleType to x
  set y to ((l^2)/((l+b))/2)
  format number y as "0"&sysdecimal&"0" from null
  set text of field "y" of group angleType to y
  set qr to ((0.67*0.67*0.707*xu)/1000)
  set vr to (qr*d)
  format number vr as "0" & sysdecimal & "000" from null
  get ("0.67X0.67X0.707X"&xu&"X"&d)
  set text of field "VVR" to it
  set text of field "VR" to ("="&vr&&"kN/mm")
  set ix to ((b*y^2)+(b/12)+(l^3/12)+(l*((l/2)-y)^2))
  format number ix as "0" from null
  get (b&"X"&y&"^2+"&b&"/12+"&l&"^3/12+"&l&"X("&l&"/2-"&y&")^2")
  set text of field "vix" to it
  set text of field "ix" to ("="&ix&&"mm^3")
  set iy to ((l*x^2)+(l/12)+(b^3/12)+(b*((b/2)-x)^2))
  format number iy as "0" from null
  get (l&"X"&x&"^2+"&l&"/12+"&b&"^3/12+"&b&"X("&b&"/2-"&x&")^2")
```

```
set text of field "viy" to it
set text of field "iy" to ("="&iy&&"mm^3")
set ip to (ix+iy)
format number ip as "0"  from null
get (ix&&"+"&&iy)
set the text of field "vip" to it
set the text of field "ip" to ("="&ip&&"mm^3")
set mf to (abs(vfx)*ly+abs(vfy)*lx)
format number mf as "0" from null
get ("¦"&vfx&"¦X"&ly&"+"&"¦"&vfy&"¦X"&lx)
set the text of field "vmf" to it
set the text of field "mf" to ("="&mf&"kN-mm")
set vx to (vfx/(l+b))
format number vx as "0" & sysdecimal&"000" from null
get (vfx&"/("&l&"+"&b&")")
put it into the text of field "vvx"
set the text of field "vx" to ("="&vx&&"kN/mm")
set vy to (vfy/(l+b))
format number vy as "0" & sysdecimal&"000" from null
get (vfy&"/("&l&"+"&b&")")
put it into the text of field "vvy"
set the text of field "vy" to ("="&vy&&"kN/mm")
if ((angleType is "ul") or (angleType is "ur")) then
   set vxb1 to (-mf*y/ip)
   get ("-"&mf&"X("&y&")/"&ip)
else
   set vxb1 to (mf*y/ip)
   get (mf&"X("&y&")/"&ip)
end
format number vxb1 as "0"&sysdecimal&"000" from null
put it into the text of field "vvxb1"
set the text of field "vxb1" to ("="&vxb1&&"kN/mm")
if ((angleType is "ul") or (angleType is "ur")) then
   set vxb2 to (-mf*(l-y)/ip)
   get ("-"&mf&"X("&l&"-"&y&")/"&ip)
else
   set vxb2 to (mf*(l-y)/ip)
   get (mf&"X("&l&"-"&y&")/"&ip)
end
format number vxb2 as "0"&sysdecimal&"000" from null
put it into the text of field "vvxb2"
set the text of field "vxb2" to ("="&vxb2&&"kN/mm")
if ((angleType is "ul") or (angleType is "ll")) then
   set vyb1 to (mf*x/ip)
   get (mf&"X"&x&"/"&ip)
else
   set vyb1 to (-mf*x/ip)
   get (mf&"X"&x&")/"&ip)
end
format number vyb1 as "0"&sysdecimal&"000" from null
put it into the text of field "vvyb1"
set the text of field "vyb1" to ("="&vyb1&&"kN/mm")
if ((angleType is "ul") or (angleType is "ll")) then
   set vyb2 to (-mf*(b-x)/ip)
   get ("-"&mf&"X("&b&"-"&x&")/"&ip)
else
```

```
    set vyb2 to (mf*(b-x)/ip)
    get (mf&"X("&b&"-"&b&")/"&ip)
  end
  format number vyb2 as "0"&sysdecimal&"000" from null
  put it into the text of field "vvyb2"
  set the text of field "vyb2" to ("="&vyb2&&"kN/mm")
  push sqrt((vx+vxb1)^2+(vy+vyb1)^2) onto va
  push sqrt((vx+vxb1)^2+(vy+vyb2)^2) onto va
  push sqrt((vx+vxb2)^2+(vy+vyb1)^2) onto va
  set big to 0
  step i from 3 to 1 by -1
    pop va
    if (it > big) then
      set big to it
      set index to i
    end if
  end step
  conditions
    when (index = 1)
      get ("((&vx&"+("&vxb1&"))^2+("&vy&"+("&vyb1&"))^2)^½)")
    when (index=2)
      get ("((&vx&"+("&vxb1&"))^2+("&vy&"+("&vyb2&"))^2)^½)")
    when (index=3)
      get ("((&vx&"+("&vxb2&"))^2+("&vy&"+("&vyb1&"))^2)^½)")
  end conditions
  set v to big
  format number v as "0"&sysdecimal&"000" from null
  put it into the text of field "vv"
  set the text of field "v" to ("="&v&&"kN/mm")


    set chob to objects of group "results"
  if (v>vr)
    while chob is not null
      pop chob
      set strokecolor of it to red
    end while
  else
    while chob is not null
      pop chob
      set strokecolor of it to blue
    end while
  end if


end
```

Script Two Horizontal Plastic

```
to handle leaveField
  local e,w,l,d


  if focus is not null
```

```
      get text of target          --localize the text of the field
      set syssuspend to false     --turn this off so no error messages
      set syserror to null        --initialize
      increment it by 1           --try a numeric operation on the text
      set syssuspend to true      --don't forget to turn this on again
      if syserror is not null     --test for an error
         request strNumericValue() with strOK() or strCancel()
         if it is strCancel()
            set syssuspendmessages to true
            set focus to null
            set syssuspendmessages to false
            break
         end
         break to system
      end
    end if
    forward leavefield
    put the text of field "e" into e
    put the text of field "w" into w
    put the text of field "l" into l
    put the text of field "d" into d
--  calculation of C by DLL
    set C to parhorc(l,w,e)
-- divive c by 1000 because the prvious function gives c*1000
    set c to (c/1000)
    set pr to (c*d*l)
    format number C as "0" & sysDecimal & "000" from null
    format number pr as "0" & sysDecimal & "0" from null
    put (c&&"X"&&d&&"X"&&l) into the text of field "Pr1"
    put ("="&&pr&&"kN") into the text of field "Pr"
    put (pr) into the text of field "Cr1"
    put (d&&"X"&&l) into the text of field "Cr2"
    put ("="&&c) into the text of field "Cr"
    put (pr) into the text of field "Dr1"
    put (c&&"X"&&l) into the text of field "Dr2"
    put ("="&&d&&"mm") into the text of field "Dr"
    put (pr) into the text of field "Lr1"
    put (c&&"X"&&d) into the text of field "Lr2"
    put ("="&&l&&"mm") into the text of field "Lr"

-- result field
    put the text of field "p" into p
    set the text of field "PRV" to (pr&&"kN")
    set the text of field "PV" to (p&&"kN")
    if (pr>=p) then
      set the text of field "signe" to ">"
      set fieldColor to "240,50,100"
    else
      set the text of field "signe" to "<"
      set fieldColor to "0,50,100"
    end
    set list to objects of group "view"
    while list is not null
      pop list
      set strokeColor of it to fieldColor
    end
```

```
end
```

Script for Box Weld Plastic

```
to handle leaveField
   local e,w,l,d


   if focus is not null
     get text of target         --localize the text of the field
     set syssuspend to false     --turn this off so no error messages
     set syserror to null        --initialize
     increment it by 1           --try a numeric operation on the text
     set syssuspend to true      --don't forget to turn this on again
     if syserror is not null     --test for an error
        request strNumericValue() with strOK() or strCancel()
        if it is strCancel()
           set syssuspendmessages to true
           set focus to null
           set syssuspendmessages to false
           break
        end
        break to system
     end
   end if
   forward leavefield
   put the text of field "e" into e
   put the text of field "w" into w
   put the text of field "l" into l
   put the text of field "d" into d
-- calculation of C by DLL
   set C to boxAllC (l,w,e)
-- divive c by 1000 because the prvious function gives c*1000
   set C to (C/1000)
   set pr to (c*d*l)
   format number C as "0" & sysDecimal & "000" from null
   format number pr as "0" & sysDecimal & "00" from null
   put (c&&"X"&&d&&"X"&&l) into the text of field "Pr1"
   put ("="&&pr&&"kN") into the text of field "Pr"
   put (pr) into the text of field "Cr1"
   put (d&&"X"&&l) into the text of field "Cr2"
   put ("="&&c) into the text of field "Cr"
   put (pr) into the text of field "Dr1"
   put (c&&"X"&&l) into the text of field "Dr2"
   put ("="&&d&&"mm") into the text of field "Dr"
   put (pr) into the text of field "Lr1"
   put (c&&"X"&&d) into the text of field "Lr2"
   put ("="&&l&&"mm") into the text of field "Lr"

-- result field
   put the text of field "p" into p
   set the text of field "PRV" to (pr&&"kN")
   set the text of field "PV" to (p&&"kN")
```

```
    if (pr>=p) then
      set the text of field "signe" to ">"
      set fieldColor to "240,50,100"
    else
      set the text of field "signe" to "<"
      set fieldColor to "0,50,100"
    end
    set list to objects of group "view"
    while list is not null
      pop list
      set strokeColor of it to fieldColor
    end

end
```

C Weld Close Side Plastic

```
to handle leaveField
    local e,w,l,d

    if focus is not null
      get text of target          --localize the text of the field
      set syssuspend to false     --turn this off so no error messages
      set syserror to null        --initialize
      increment it by 1           --try a numeric operation on the text
      set syssuspend to true      --don't forget to turn this on again
      if syserror is not null     --test for an error
         request strNumericValue() with strOK() or strCancel()
         if it is strCancel()
            set syssuspendmessages to true
            set focus to null
            set syssuspendmessages to false
            break
         end
         break to system
      end
    end if
    forward leavefield
    put the text of field "e" into e
    put the text of field "w" into w
    put the text of field "l" into l
    put the text of field "d" into d
    set xl to (w^2/(l+2*w))
    format number xl as "0" & sysdecimal & "0" from null
    put xl into the text of field "k"
--  calculation of C by DLL
    set C to cweldcc (l,w,e)
-- divive c by 1000 because the prvious function gives c*1000
    set C to (C/1000)
    set pr to (c*d*l)
    format number C as "0" & sysDecimal & "000" from null
    format number pr as "0" & sysDecimal & "0" from null
    put (c&&"X"&&d&&"X"&&l) into the text of field "Pr1"
    put ("="&&pr&&"kN") into the text of field "Pr"
```

```
    put (pr) into the text of field "Cr1"
    put (d&&"X"&&l) into the text of field "Cr2"
    put ("="&&c) into the text of field "Cr"
    put (pr) into the text of field "Dr1"
    put (c&&"X"&&l) into the text of field "Dr2"
    put ("="&&d&&"mm") into the text of field "Dr"
    put (pr) into the text of field "Lr1"
    put (c&&"X"&&d) into the text of field "Lr2"
    put ("="&&l&&"mm") into the text of field "Lr"

-- result field
    put the text of field "p" into p
    set the text of field "PRV" to (pr&&"kN")
    set the text of field "PV" to (p&&"kN")
    if (pr>=p) then
      set the text of field "signe" to ">"
      set fieldColor to "240,50,100"
    else
      set the text of field "signe" to "<"
      set fieldColor to "0,50,100"
    end
    set list to objects of group "view"
    while list is not null
      pop list
      set strokeColor of it to fieldColor
    end


end
```

Script C Open Plastic

```
to handle leaveField
   local e,w,l,d


   if focus is not null
     get text of target          --localize the text of the field
     set syssuspend to false     --turn this off so no error messages
     set syserror to null        --initialize
     increment it by 1           --try a numeric operation on the text
     set syssuspend to true      --don't forget to turn this on again
     if syserror is not null     --test for an error
        request strNumericValue() with strOK() or strCancel()
        if it is strCancel()
           set syssuspendmessages to true
           set focus to null
           set syssuspendmessages to false
           break
        end
        break to system
     end
   end if
   forward leavefield
```

```
    put the text of field "e" into e
    put the text of field "w" into w
    put the text of field "l" into l
    put the text of field "d" into d
    set xl to (w^2/(l+2*w))
    format number xl as "0" & sysdecimal & "0" from null
    put xl into the text of field "k"
--  calculation of C by DLL
    set C to cweldoc (l,w,e)
-- divive c by 1000 because the prvious function gives c*1000
    set C to (C/1000)
    set pr to (c*d*l)
    format number C as "0" & sysDecimal & "000" from null
    format number pr as "0" & sysDecimal & "0" from null
    put (c&&"X"&&d&&"X"&&l) into the text of field "Pr1"
    put ("="&&pr&&"kN") into the text of field "Pr"
    put (pr) into the text of field "Cr1"
    put (d&&"X"&&l) into the text of field "Cr2"
    put ("="&&c) into the text of field "Cr"
    put (pr) into the text of field "Dr1"
    put (c&&"X"&&l) into the text of field "Dr2"
    put ("="&&d&&"mm") into the text of field "Dr"
    put (pr) into the text of field "Lr1"
    put (c&&"X"&&d) into the text of field "Lr2"
    put ("="&&l&&"mm") into the text of field "Lr"

-- result field
    put the text of field "p" into p
    set the text of field "PRV" to (pr&&"kN")
    set the text of field "PV" to (p&&"kN")
    if (pr>=p) then
      set the text of field "signe" to ">"
      set fieldColor to "240,50,100"
    else
      set the text of field "signe" to "<"
      set fieldColor to "0,50,100"
    end
    set list to objects of group "view"
    while list is not null
      pop list
      set strokeColor of it to fieldColor
    end


end
```

Script L Close Plastic

```
to handle leaveField
   local e,w,l,d

   if focus is not null
     get text of target          --localize the text of the field
     set syssuspend to false     --turn this off so no error messages
```

```
    set syserror to null        --initialize
    increment it by 1           --try a numeric operation on the text
    set syssuspend to true      --don't forget to turn this on again
    if syserror is not null     --test for an error
       request strNumericValue() with strOK() or strCancel()
       if it is strCancel()
          set syssuspendmessages to true
          set focus to null
          set syssuspendmessages to false
          break
       end
       break to system
    end
  end if
  forward leavefield
  put the text of field "e" into e
  put the text of field "w" into w
  put the text of field "l" into l
  put the text of field "d" into d
  set x1 to ((0.5*w^2)/(l+w))
  set x2 to ((0.5*l^2)/(l+w))
  format number x1 as "0" & sysdecimal & "0" from null
  format number x2 as "0" & sysdecimal & "0" from null
  put x1 into the text of field "kx"
  put x2 into the text of field "ky"
--  calculation of C by DLL
  set C to lweldcc (l,w,e)
-- divive c by 1000 because the prvious function gives c*1000
  set C to (C/1000)
  set pr to (c*d*l)
  format number C as "0" & sysDecimal & "000" from null
  format number pr as "0" & sysDecimal & "0" from null
  put (c&&"X"&&d&&"X"&&l) into the text of field "Pr1"
  put ("="&&pr&&"kN") into the text of field "Pr"
  put (pr) into the text of field "Cr1"
  put (d&&"X"&&l) into the text of field "Cr2"
  put ("="&&c) into the text of field "Cr"
  put (pr) into the text of field "Dr1"
  put (c&&"X"&&l) into the text of field "Dr2"
  put ("="&&d&&"mm") into the text of field "Dr"
  put (pr) into the text of field "Lr1"
  put (c&&"X"&&d) into the text of field "Lr2"
  put ("="&&l&&"mm") into the text of field "Lr"

-- result field
  put the text of field "p" into p
  set the text of field "PRV" to (pr&&"kN")
  set the text of field "PV" to (p&&"kN")
  if (pr>=p) then
    set the text of field "signe" to ">"
    set fieldColor to "240,50,100"
  else
    set the text of field "signe" to "<"
    set fieldColor to "0,50,100"
  end
  set list to objects of group "view"
```

```
    while list is not null
      pop list
      set strokeColor of it to fieldColor
    end


end
```

Script L Weld Open Plastic

```
to handle leaveField
    local e,w,l,d


    if focus is not null
      get text of target          --localize the text of the field
      set syssuspend to false     --turn this off so no error messages
      set syserror to null        --initialize
      increment it by 1           --try a numeric operation on the text
      set syssuspend to true      --don't forget to turn this on again
      if syserror is not null     --test for an error
         request strNumericValue() with strOK() or strCancel()
         if it is strCancel()
            set syssuspendmessages to true
            set focus to null
            set syssuspendmessages to false
            break
         end
         break to system
      end
    end if
    forward leavefield
    put the text of field "e" into e
    put the text of field "w" into w
    put the text of field "l" into l
    put the text of field "d" into d
    set x1 to ((0.5*w^2)/(l+w))
    set x2 to ((0.5*l^2)/(l+w))
    format number x1 as "0" & sysdecimal & "0" from null
    format number x2 as "0" & sysdecimal & "0" from null
    put x1 into the text of field "kx"
    put x2 into the text of field "ky"
--  calculation of C by DLL
    set C to lweldoc (l,w,e)
-- divive c by 1000 because the prvious function gives c*1000
    set C to (C/1000)
    set pr to (c*d*l)
    format number C as "0" & sysDecimal & "000" from null
    format number pr as "0" & sysDecimal & "0" from null
    put (c&&"X"&&d&&"X"&&l) into the text of field "Pr1"
    put ("="&&pr&&"kN") into the text of field "Pr"
    put (pr) into the text of field "Cr1"
    put (d&&"X"&&l) into the text of field "Cr2"
```

```
    put ("="&&c) into the text of field "Cr"
    put (pr) into the text of field "Dr1"
    put (c&&"X"&&l) into the text of field "Dr2"
    put ("="&&d&&"mm") into the text of field "Dr"
    put (pr) into the text of field "Lr1"
    put (c&&"X"&&d) into the text of field "Lr2"
    put ("="&&l&&"mm") into the text of field "Lr"

-- result field
    put the text of field "p" into p
    set the text of field "PRV" to (pr&&"kN")
    set the text of field "PV" to (p&&"kN")
    if (pr>=p) then
      set the text of field "signe" to ">"
      set fieldColor to "240,50,100"
    else
      set the text of field "signe" to "<"
      set fieldColor to "0,50,100"
    end
    set list to objects of group "view"
    while list is not null
      pop list
      set strokeColor of it to fieldColor
    end


end
```

Script Flexible Beam to Column Connection

```
to handle buttonDown

  -- this button's script controls the print in the record Field
  -- of the background results


  local a,ptext

  set syscursor to 4

  get text of field "ptext1" of this page
  push it onto ptext
  push "Beam end shear reaction:" onto ptext
  get text of field "reaction"
  set R to it
  get "R"&TAB&"= beam end shear force"&TAB&"="&TAB&it&"kN"&TAB&"<input<"
  push it onto ptext
  get text of field "bd"
  set bd to it
  get "bd"&TAB&"= beam depth        "&TAB&"="&TAB&it&"mm"&TAB&"<input<"
  push it onto ptext
  get text of field "bfw"
  set bfw to it
  get "bfw"&TAB&"= beam flange width"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
```

```
push it onto ptext
get text of field "bft"
set bft to it
get "bft"&TAB&"= beam flange thickness"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
get text of field "bwt"
set bwt to it
get "bwt"&TAB&"= beam web thickness"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
get text of field "bkv"
set bkv to it
get "k"&TAB&"= beam k value      "&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
get text of field "bys"
set bys to it
get "Fyb"&TAB&"= beam minimum yield stress"&TAB&"="&TAB&it&"MPa"&TAB&"<input<"
push it onto ptext
get text of field "bmf"
set bmf to it
get "phi"&TAB&"= beam material resis. factor"&TAB&"="&TAB&it&""&TAB&"<input<"
push it onto ptext
push CRLF onto ptext
push "Column paramaters:" onto ptext
get text of field "cfw"
set cfw to it
get "cfw"&TAB&"= column flange width"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
get text of field "cft"
set cft to it
get "cft"&TAB&"= column flange thickness"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
get text of field "cys"
set cys to it
get "Fyc"&TAB&"= column minimum yield stress"&TAB&"="&TAB&it&"MPa"&TAB&"<input<"
push it onto ptext
push CRLF onto ptext
push "Electrode parameters:" onto ptext
get typeElec of button "electrode" of this page
set xu to it
get "Xu"&TAB&"= elctrode ultimate strength"&TAB&"="&TAB&it&"MPa"&TAB&"<input<"
push it onto ptext
push CRLF onto ptext
push "Web crippling: Minimum length of bearing required" onto ptext
set nn to (((R*1000)/(bys*1.1*bmf*bwt))-(2.5*bkv))
format number NN as "0" & sysdecimal &"0" from null
get "NN"&TAB&"= (R*1000)/(Fyb*1.1*phi*bwt)-2.5*k"&TAB&"="&TAB & nn &"mm"&TAB&"<calc<"
push it onto ptext
push CRLF onto ptext
push "Minimum width of seat for bearring:" onto ptext
set nws to (nn+20)
format number NWS as "0" & sysdecimal &"0" from null
get "NWS"&TAB&"= NN+20              "&TAB&"="&TAB & NWS &"mm"&TAB&"<calc<"
push it onto ptext
push CRLF onto ptext
push "Angle parameters:"  onto ptext
get text of field "ays"
```

```
set ays to it
get "Fya"&TAB&"= angle minimum yield stress"&TAB&"="&TAB&it&"MPa"&TAB&"<input<"
push it onto ptext
get text of field "alos"
set alos to it
get "als"&TAB&"= leg length: outstanding seat"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
get text of field "alc"
set alc to it
get "all"&TAB&"= leg length: welded to column"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
get text of field "ath"
set ath to it
get "ath"&TAB&"= angle thickness   "&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
push CRLF onto ptext
push "Weld fastening parameters:" onto ptext
get text of field "fsb"
set fsb to it
get "fsb"&TAB&"= fillet size: beam to angle"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
conditions
  when (bft>20)
    set it to 8
  when ((bft<=20) and (bft>=12))
    set it to 6
  when (bft<12)
    set it to 5
end conditions
if (fsb>=it) then
   set it to "ok"
else
   set it to "too small"
end
push CRLF onto ptext
get TAB&"Check minimum size fillet on "&quote&"bft"&quote&""&TAB&it&"   W59 4.3.4.2"
push it onto ptext
if (bft<6) then
  set it to bft
else
  set it to (bft-2)
end
if (fsb<=it) then
  set it to "ok"
else
  set it to "too big"
end
get TAB&"Check maximum size fillet on "&quote&"bft"&quote&""&TAB&it&"   W59 4.4.1.2"
push it onto ptext
push CRLF onto ptext
get text of field "fsc"
set fsc to it
get "fsc"&TAB&"= fillet size: column to angle"&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
conditions
  when (ath>20)
```

```
    set it to 8
  when ((ath<=20) and (ath>=12))
    set it to 6
  when (ath<12)
    set it to 5
end conditions
if (fsc>=it) then
   set it to "ok"
else
   set it to "too small"
end
push CRLF onto ptext
get TAB&"Check minimum size fillet on "&quote&"ath"&quote&""&TAB&it&"   W59 4.3.4.2"
push it onto ptext
if (ath<6) then
  set it to ath
else
  set it to (ath-2)
end
if (fsc<=it) then
  set it to "ok"
else
  set it to "too big"
end
get TAB&"Check maximum size fillet on "&quote&"ath"&quote&""&TAB&it&"   W59 4.4.1.2"
push it onto ptext
push CRLF onto ptext
get text of field "ptext2" of this page
push it onto ptext
push CRLF onto ptext
push "Determine eccentricty of reaction to critical angle x-section" onto ptext
set eccen to ((alos-20)/2+10-ath)
get "eccen"&TAB&"= ((als-20)/2+10-ath)"&TAB&"="&TAB & eccen &"mm"&TAB&"<calc<"
push it onto ptext
push "Determine minimum angle length: based on full yield in angle leg x-sect." onto ptext
set MIN1 to ((R*1000*4*eccen)/(bmf*ays*(ath^2)))
format number min1 as "0" & sysdecimal &"0" from null
get "MIN1"&TAB&"=(R*4000*eccen/bmf*ays*(ath^2))"&TAB&"="&TAB & MIN1 &"mm"&TAB&"<calc<"
push it onto ptext
push "Mimimun recommended edge distance for fillet welds on column flange:" onto ptext
set edcf to ((7*fsc)/6+7)
format number edcf as "0" from null
get "edcf"&TAB&"= ((7*fsc)/6+7)      "&TAB&"="&TAB & edcf &"mm"&TAB&"<calc<"
push it onto ptext
push "Maximum length of seat angle to still permit welding to column" onto ptext
set maxl to (cfw-2*edcf)
get "maxl"&TAB&"= (cfw-2*edcf)         "&TAB&"="&TAB & maxl &"mm"&TAB&"<calc<"
push it onto ptext
push "Mimimun recommended edge distance for fillet welds on seat angle:" onto ptext
set edsa to ((7*fsb)/6+7)
format number edsa as "0" from null
get "edsa"&TAB&"= ((7*fsb)/6+7)         "&TAB&"="&TAB & edsa &"mm"&TAB&"<calc<"
push it onto ptext
push "Minimum length of seat angle to still permit welding to beam flange" onto ptext
set MIN2 to (bfw+2*edsa)
get "MIN2"&TAB&"= (bfw+2*edsa)       "&TAB&"="&TAB & MIN2 &"mm"&TAB&"<calc<"
```

```
push it onto ptext
push CRLF onto ptext
get text of field "ptext3"
push it onto ptext
push CRLF onto ptext
push "Chosen length of angle seat:" onto ptext
get text of field "alas"
set alas to it
get "alas"&TAB&"= angle length      "&TAB&"="&TAB&it&"mm"&TAB&"<input<"
push it onto ptext
push CRLF onto ptext
push  text of field "ptext4" onto ptext
push CRLF onto ptext
push "Factored resistance of angle to column welds per mm of weld length:" onto ptext
push  "base metal..." onto ptext
get min(cys,ays)
set fy to it
get "Fy"&TAB&"= min yield stress"&TAB&"="&TAB&it&"MPa"&TAB&"<calc<"
push it onto ptext
set V1r to ((0.67*0.9*fy*fsc)/1000)
format number V1r as "0" & sysdecimal &"000" from null
get "V1r"&TAB&"= ((0.67*0.9*fy*fsc)/1000)"&TAB&"="&TAB&V1r&"kN/mm"&TAB&"<calc<"
push it onto ptext
push "weld metal..." onto ptext
set V2r to (0.67*0.67*(fsc/(2^0.5))*Xu/1000)
format number V2r as "0" & sysdecimal &"000" from null
get "V2r"&TAB&"= 0.67*0.67*(fsc/(2^0.5))*Xu/1000"&TAB&"="&TAB&V2r&"kN/mm"&TAB&"<calc<"
push it onto ptext
push CRLF onto ptext
push "Minimum resistance..." onto ptext
get min(V1r,V2r)
set Vr to it
get "Vr"&TAB&"= minimum resistance"&TAB&"="&TAB&Vr&"kN/mm"&TAB&"<calc<"
push it onto ptext
push "Minimum column flange thickness: " onto ptext
set mcft to ((Vr*1000)/(0.66*0.9*cys))
format number mcft as "0" & sysdecimal &"0" from null
get "mcft"&TAB&"= ((Vr*1000)/(0.66*0.9*cys))"&TAB&"="&TAB&mcft&"mm"&TAB&"<calc<"
push it onto ptext
push CRLF onto ptext
push text of field "ptext5" onto ptext
push CRLF onto ptext
push " instantaneous center coefficient data "&CRLF onto ptext
get "Angle to column weld:"&CRLF&"Weld geometry..."
push it onto ptext
get "ll"&TAB&"= vert. length of column welds"&TAB&"="&TAB & alc &"mm"&TAB&"<calc<"
push it onto ptext
set ee to (eccen+ath+10)
format number ee as "0" from null
get "ee"&TAB&"= (eccen+ath+10)     "&TAB&"="&TAB & ee &"mm"&TAB&"<calc<"
push it onto ptext
get "Fy"&TAB&"= Min. yield stress"&TAB&"="&TAB & fy &"MPa"&TAB&"<calc<"
push it onto ptext
set abl to text of field "abl"
get "abl"&TAB&"= Angle bearing length"&TAB&"="&TAB & abl &"mm"&TAB&"<calc<"
push it onto ptext
```

```
  get "fsc"&TAB&"= Fillet weld size    "&TAB&"="&TAB & fsc &"mm"&TAB&"<calc<"
  push it onto ptext
  push CRLF onto ptext
  get outplaneverc(alas,ee,fy,abl,fsc)
  set C to it
 -- divide c by 1000 because function returns c*1000
  set c to (c/1000)
  format number C as "0" & sysDecimal & "000" from null
  get "C"&TAB&"= Coefficient          "&TAB&"="&TAB & C &"kN/mm^2"&TAB&"<calc<"
  push it onto ptext
  push CRLF onto it
  push "Factored capacity of column flange welds:" onto ptext
  set fccf to (alc*C)
  format number fccf as "0" & sysdecimal & "0" from null
  get "fccf"&TAB&"= (ll * C)              "&TAB&"="&TAB & fccf &"kN"&TAB&"<calc<"
  push it onto ptext
  push "Connection reduction factor due to column flange thickness..." onto ptext
  if (cft<mcft) then
    set crfc to (cft/mcft)
    format number crfc as "0" & sysdecimal & "00" from null
  else
    set crfc to 1
  end
  get "crfc"&TAB&"= reduction factor        "&TAB&"="&TAB & crfc &""&TAB&"<calc<"
  push it onto ptext
  push "Reduce double angle to column flange capacity..." onto ptext
  set cap to (fccf*crfc)
  format number ee as "0"&sysdecimal&"0" from null
  get "cap"&TAB&"= (fccf*crfc)          "&TAB&"="&TAB & cap &"kN"&TAB&"<calc<"
  push it onto ptext

  push CRLF onto ptext
  push CRLF onto ptext

  if (cap>R) then
      push ("Welded connection's capacity is sufficiant:"&&cap&"kN >"&&R&&"kN") onto ptext
  else
      push ("Welded connection's capacity is not sufficiant:"&&cap&"kN <"&&R&&"kN") onto ptext
  end



  set a to  reverseList (ptext)
  set the text of recordfield "viewR" of page "results" to null
  while a is not null
    pop a
    put it after the  text of recordfield "viewR" of page "results"
    put CRLF after the text of recordfield "viewR" of page "results"
  end
  go to page "results"

  set syscursor to default


end
```

```
to get reverseList stack
  local tempStack
  put null into tempStack
  while stack <> null
    pop stack
    push it onto tempStack
  end while
  return tempStack
end
```

Script For Plate Distrotion Page

```
to handle enterpage
  local i
  step i from 1 to 14
  set it to ("paintobject pl"&i&&"of page plate_distortion")
  hide it
  end step
  show paintobject "pl1" of page plate_distortion
  step i from 1 to 3
     set it to "field champs"&i&&" of page plate_distortion"
     set activated of it to true
     hide it
   end step
   -- can't go to next page
   send disable to button "next" of this background
end

to handle leavePage
   send enable to button "next" of this background
end
```

Script Apply Weld Button

```
to handle buttonUp
  hide paintobject "pl14" of page plate_distortion
  hide paintobject "pl13" of page plate_distortion
  hide paintobject "pl12" of page plate_distortion
  show field champs1 of page plate_distortion
  show paintobject "pl1" of page plate_distortion
  show paintobject "soudeur" of page plate_distortion
  step i from 2 to 12
    set it to " paintobject pl"&i&&"of page plate_distortion"
    show  it
    move paintobject "soudeur" of page plate_distortion to (1720+i*250),(1600-100*i)
    pause 10 ticks
  end step
--  step i from 1 to 11
--    set it to "pl"&i
--    hide paintobject it
```

```
-- end step
  hide paintobject "soudeur" of page plate_distortion

end buttonUp
```

Script for Cool Down Button

```
to handle buttonup
    step i from 1 to 12
      set it to "pl"&i
      hide paintobject it
    end step
    show paintobject "pl13"
    hide field "champs1"
    show field "champs2"
end
```

Script for Cold Button

```
to handle buttonUP
  hide paintobject "pl13"
  show paintobject  "pl14"
  hide field "champs2"
  show field "champs3"
end
```

Konwledge Base

```
to get helpVersion
   return "WISDOM 92r Info"
end

to handle quit
  send save
  send exit
end


to get FailGo what, where
   set sysSuspend to false
   set sysError to null
   set sysCursor to 4
   if where contains "Def"
      go page (characters 1 to 32 of (what && "(Def.)") )
   else
      go page (characters 1 to 32 of what)
   end
   get sysError
   set sysSuspend to true
   set sysCursor to default
   return it <> null
end

to get searchBackground bkgnd, startp
   return searchBackground(bkgnd, startp) of page 2
end


to handle KeyChar x
   system FixRef
   if FixRef <> null and x = chartoansi("#") then
      set pN to name of this page
      go page (word 1 of FixRef)
      clear word 1 of FixRef
      set isText to not (pN contains "(Def.)")
      if text of recordfield "Hotwords" <> null then
         put CRLF after text of recordfield "Hotwords"
      end if
      if isText then
         put FixRef & "~@" & pN after text of recordfield "Hotwords"
      else
         clear last word of pN
         put FixRef & "~" & pN after text of recordfield "Hotwords"
      end if
      set FixRef to null
   else
      forward to system
   end if
end keyChar
```

```
to handle EnterBook
  system FixRef, Testing, windowSized, s_authorOK

  set syslockscreen to true
    restore menuBar at both
    remove menu "file" at reader
    remove menu "edit" at reader
    remove menu "text" at reader
    remove menu "help" at reader
    add menu "&File"  at reader
          add menuItem "&Compact Book" to menu "&File" at reader
    add menuItem "&Quit"  to menu "&File" at reader
  remove menu "page" at reader
  set sysSuspend to true
  set syscursor to 4
  send reader
  set sysFontFace to arial
  set sysFontSize to 10
  set sysFontStyle to bold
  set sysDecimal to "."
  set sysDateFormat to "m/d/y"
  set s_authorOK to false
  hide scrollbar
  forward
  set Testing to false
  set sysChangesDB to false
  set FixRef to null
  if windowSized is null then
    send sizeToPage
    set windowSized to true
  end if
  set syslockscreen to false
  send setHelpMenus to page 1
  send fixHotWordMenu
  send setSuffixes
  send setPrefixes
  set syscursor to 1
end

to handle doInfo contextInfo

    set syscoursor to 4

    if argcount = 0 then
       send IndexofTopics
       set syscursor to default
       break doinfo
    end if

    set HlpTopic to trim (item 1 of contextInfo)

    while HlpTopic is not null

        if not (failGO(HlpTopic) and FailGo(HlpTopic, "def"))
           -- found the page
           break while
```

```
        end

        set wList to text of field "context" of page 2
        set L to 0
        set wL to null
        step L from 1 to textlineCount(wlist)
           set wL to textLine L of wList
           if offset (HlpTopic,wL)=1 then
              if not failGo( trim (item 2 of wl) )
                 break while
              end
           end
        end step


        --c'ant identify
        send IndexofTopics
        break while
    end while
    send showmywindow
    set syscursor to default
end

to handle setSuffixes
        system s_suffixes

        clear s_suffixes
        get "s"
        while it is not null
                pop it after s_suffixes
                put CRLF after s_suffixes
        end
end

to handle setprefixes
        system s_prefixes

        clear s_prefixes
        get ""
        while it is not null
                pop it after s_prefixes
                put CRLF after s_prefixes
        end
end

to handle fixHotWordMenu
   send fixHotWordMenu to page 2
end

to handle testHotWords pStart, pEnd
   send testHotWords pStart, pEnd to page 1
end

to handle LeaveBook
   restore menubar at both
   restore system
```

```
end

to handle showMyWindow
   send showmywindow to page 2
end


to handle author
   send author to page 1
end

to handle reader
   if sysLevel is "Author"
      set sysLevel to reader
      send sizeToPage
      forward
   end
end

to handle help
    get group "help" of this background
    show it
    set focus to button "OK" of it
end




-- Handle F3 key at Reader level (Author not in menu)
to handle keyDown x
   if x = keyF3
      send Author
   else
      forward
   end
end

to handle buttonUp
   if word 1 of target is "hotword" then
      get text of target
      send doHotWord (it)
   end if
end

to get HotL what
   set wList to text of recordfield "Hotwords"
   set L to 0
   set wL to null
   set wrd to what & "~"
   while L < textlinecount(wList) and wL is null
      increment L
      if offset(wrd, textline L of wList) = 1 then
         set wL to textline L of wList
```

```
      end if
   end while
   return wL
end

to handle seekTopic what
   send seekTopic what to page 2
end

to handle doHotWord what
   send doHotWord what to page 2
end

to handle fix WrdTxt
   send fix WrdTxt,(pageNumber of this page) to page 1
end

to get trim x
   set trimSet to " " & quote & tab
   while x<> null and first char of x is in trimSet
      clear first char of x
   end
   while x <> null and last char of x is in trimSet
      clear last char of x
   end
   return x
end


to handle IndexofTopics
   go to first  page of background  "topic index"
end

to handle searchAllText
   send searchBook to page 1
end


to handle SearchIndexOfTopics
   send SearchIndexOfTopics to background "Topic Index"
end




to handle AlwaysShown
   set my HotWordShow to "Show"
   send FixHotwordMenu
end

to handle AlwaysHidden
   set my HotWordShow to "Hide"
   send FixHotwordMenu
end
```

```
to handle ShowWhenMouseIsInWindow
   set my HotWordShow to null
   send FixHotwordMenu
end

to handle idle
   if my HotWordShow is null then
     get sysMousePosition
     set hws to (item 1 of it >= 0 \
       and item 1 of it <= item 1 of my size \
       and item 2 of it >= 0 \
       and item 2 of it <= item 2 of my size)
     if hws <> sysHotwordsShown
          set sysHotwordsShown to hws
     end
   end
   forward
end

to handle HotwordList
   send HotwordList to page 1
end


-- Filters each char in c from string x
to get filter c, x
   step i from 1 to charCount(c)
      get char i of c
      do
         set p to offset (it, x)
         if p > 0
            clear char p of x
         end
      until p <= 0
   end
   return x
end

-- Names the current page
to handle namePage x
   get filter (quote & leftQuote & rightQuote, x)
   set name of this page to chars 1 to 32 of it
end

-- functions
-- strip a CRLF from a string
to get stripCRLF str
   get offset(CRLF, str)
   if it <> 0 then
     clear chars it to it + 1 of str
   end
   return str
end

to handle sortrecord1
   set a to text of recordfield "topics"
```

```
    set b to 1
    set it to insertionsort (a,b)
    put it into text of recordfield "topics"
 end



 to get INSERTIONSORT  fTable, fKeyNum

    local  vLineCount,     --the number of lines in fTable \
           vRecordNumber,  --the line number being inserted \
           vCurKey,        --the current key value \
           vSortedCount,   --the number of records already sorted \
           vTemp           --holder for the line being inserted during its deletion

    set vLineCount to textlinecount(fTable)

    conditions
    when vLineCount = 0
       set sysError to "Empty table"
       return null
    when vLineCount = 1
       return fTable   --it's already sorted!
    else

       step vRecordNumber from 2 to vLineCount
          if fKeyNum > itemcount (textline vRecordNumber of fTable)
             set sysError to "Key number greater than number of elements"
             return fTable
          end
          set vCurKey to item fKeyNum of textline vRecordNumber of fTable
          set vSortedCount to vRecordNumber-1
          while   vSortedCount > 0 and \
                vCurKey < item fKeyNum of textline vSortedCount \
                   of fTable as text
             decrement vSortedCount
          end
          set vTemp to textline vRecordNumber of fTable
          clear textline vRecordNumber of fTable
          if vSortedCount<>vLineCount-1
             put vTemp & CRLF before textline vSortedCount+1 of fTable
          else
             put vTemp into textline vSortedCount+1 of fTable
          end
       end
       return fTable
    end
 end



 -- strings for messages

 to get strValidDate
```

```
    return "Please enter a valid date."
end

to get strCancel
    return "Cancel"
end

to get strOK
    return "OK"
end

to get strExeNotInPath exe
    return "The directory for" && exe && "is not in your current path.  Change your path to include" && exe
& "'s directory."
end

to get strNoPicture
    return "Can't find the Figures. Please copy WPICTURE.TBK into this directory."
end

to get strNoWisdom
    return "Can't find Wisdom'92 . Please copy WISDOM.TBK into this directory."
end


to get strCantFind filename
   return "Can't find "&&filename&&"."
end



--squeeze the hell out of a modified book
--don't forget to change to the correct name of this book!
to handle compactBook
  get name of this book
  if it is null
     request "Can't compact a untitled book."
     break to system
  end if
  clear sysError
  save as "X.TBK",true
  if sysError is not null
      request "Save as X.TBK failed:"&&sysError
      break to system
  end if
  save as it ,true
  if sysError is not null
      request "Save as original name failed:"&&sysError
      break to system
  end if
end compactBook
```

Script For Page 1 of Knowledge Base

```
-- script for page 1

to handle buttonUp
   go next page
end buttonUp

to handle keyChar
   go next page
end keyChar

to handle enterpage
   set sysTimeFormat to "seconds"
   set my startT to sysTime
--   if syslevel is reader
--      show field "Credits"
--   end
end

to handle leavePage
         set sysLockScreen to true
--   hide field "Credits"
end

to handle idle
   -- don't go to main menu immediately, in case DDE command coming
   set sysTimeFormat to "seconds"
   if sysTime - my startT > 3 then
      go page "cover page"
   end
   forward
end idle

to handle author
         system s_authorOK

   -- If this isn't the first time to author level this session
   if not s_authorOK
      request "Even minor changes at Author level may keep this book from" && "working correctly.  Go
to author level anyway?" with \
         "No" or "Yes"
      if it is "No"
         break to system
      else
         set s_authorOK to true
      end
   end

   set sysLevel to author
   set sysChangesDB to true
   show commandWindow
   send sizeToPage
   forward to system
end
```

```
to handle setHelpMenus
  add menu "&Search" at both
 add menuItem "Search &Index of Topics..." alias "SearchIndexOfTopics" \
   to menu "Search" at both
  add menuItem "Search A&ll Text..." alias "SearchAllText" \
   to menu "Search" at both
  add menu "Hotwor&ds" at both
  add menuItem "Show When &Mouse Is In Window" alias "ShowWhenMouseIsInWindow" \
   to menu "Hotwords" at both
  add menuItem "Always &Shown" alias "AlwaysShown" to menu "Hotwords" at both
  add menuItem "Always &Hidden" alias "AlwaysHidden" to menu "Hotwords" at both
end

to handle fix WrdTxt, pageN --**
   system FixRef
   request "No match for" && quote & WrdTxt & quote\
    & ". Redirect to another topic?" with "Yes" or "Cancel"
   if it is "Cancel" then
      break fix
   end
   request "Go to the topic.  When ready, press" && quote \
      & "#" & quote
   set FixRef to (pageN && WrdTxt)
end

to handle TestHotWords pStart, pEnd -- Test hotwords --**
   system Testing, HotCount, BadCount, lastTPage, bFN
   set oldTimeFormat to sysTimeFormat
   set sysTimeFormat to "h:min:sec AMPM"
   set Testing to true
   set HotCount to 0
   set BadCount to 0
   set lastTPage to null
   set syslockscreen to true
   set bFN to "BADWORDS.LST"
   createfile bFN
   writefile helpVersion() & CRLF to bFN
   writefile "Test started on" && sysDate && "at" && systime & CRLF to bFN
   set sysSuspendMessages to true
         if pStart is null
                  set pStart to 18
                  set pEnd to pageCount of this book
         else
                  if pEnd is null
                           set pEnd to pStart
                  end
         end
   step i from pStart to pEnd
      put "Checking page" && i && "of" && pEnd \
         & "; found" && HotCount && "hotwords so far;"\
         && BadCount && "bad."
      go page i
      get objects of parent of this page
      set rFields to null
      while it <> null
```

```
            pop it into obj
            if object of obj is recordfield then
               push "recordfield id" && idNumber of obj onto rFields
            else
               if object of obj is "group" then
                  set objs to objects of obj
                  while objs <> null
                     pop objs into subObj
                     if object of subObj is recordfield then
                        push "recordfield id" \
                        && idNumber of subObj onto rFields
                     end if
                  end
               end if
            end
         end
         get objects of page i
         while it <> null
            pop it into obj
            if obj contains "field" then
               push obj onto rFields
            end
         end
         while rFields <> null
            pop rFields into thisField
            get objects of thisField
            while it <> null
               pop it into hWrd
               increment HotCount
               set oldp to this page
               send buttonUp to hWrd
               go oldp
            end
         end
      end
      set sysSuspendMessages to false
      writefile "---End of list:" && HotCount && "hotwords tested,"\
          && BadCount && "bad." & CRLF & sysTime & CRLF to bFN
      closefile bFN
      put "Done:" && HotCount && "hotwords tested;" && Badcount && "bad."
      set HotCount to null
      set BadCount to null
      set testing to false
      set sysTimeFormat to oldTimeFormat
   end

   to handle logBadWord what --**
      system lastTPage, BadCount, bFN
      if lastTPage <> (this page) then
         writeFile "----------------------------" \
            & CRLF & "Page" && pagenumber of this page \
            & ":" && name of this page & CRLF to bFN
      end if
      writeFile "[" & what & "]" & CRLF to bFN
         put what
      set lastTPage to (this page)
```

```
   increment BadCount
end

to handle HotwordList
   get recordfield "hotwords"
   set visible of it to not visible of it
   if visible of recordfield "hotwords" then
      check menuItem "HotwordList" at author
   else
      uncheck menuItem "HotwordList" at author
   end if
end Hotwordlist

to handle trimEnds
   get this background
   set pCount to pagecount of it
   step i from 1 to pCount
      set f to recordfield "extra" of page i of it
      while chartoansi(last character of text of f) < 33\
            and text of f <> null
         clear last character of text of f
      end
      put pCount - i
   end
end

to handle SearchBook
   system SearchString, FoundFocus, FoundPage
   if arglist is null and sysLevel is reader then
      set sysError to null
      ask "Search all book pages for:" with SearchString
      if sysError is cancel or it is null then
         break to system
      end
      set SearchString to it
      set oldP to this page
      set syslockscreen to true
      go next page
      search for it
      if selectedTextState is null
         set msg1 to "Cannot find" && quote & it & quote & "."
         if pageNumber of oldP > 1
            request msg1 \
              && "Continue search from the beginning of the book?"\
              with "Yes" or "No"
            if it is "Yes"
               go page 2
               search for it
               if selectedTextState is null
                  request msg1
                  go oldP
               end
            else
               go oldP
            end
         else
```

```
            go oldP
         end
      end
   else
      forward
   end
end
```

Script For Page 2 Of Knowledge Base

```
-- script for page 2

to handle enterpage
   add menu "&Context" at author
   add menuItem "Context Info Table..." alias "ContextInfoTable" \
      to menu "&Context" at author
   forward
end

to handle leavepage
   set sysSuspend to false
   remove menu "&Context" at author
   hide field "Context"
   set sysSuspend to true
   forward
end

to handle ContextInfoTable
   get the visible of field "Context"
   set the visible of field "Context" to not it
end

to handle ShowMyWindow
   linkDLL user
      int IsIconic(word)
      int ShowWindow(Word, int)
      int SetActiveWindow(Word)
   end linkDLL
   if isIconic(sysWindowHandle) > 0 then
      get showWindow(sysWindowHandle, 1)
   end if
   get SetActiveWindow(sysWindowHandle)
   unlinkDLL "user"
end

to handle fixHotWordMenu
   uncheck menuItem "Always Shown"
   uncheck menuItem "Always Hidden"
   uncheck menuItem "Show When Mouse Is In Window"
   conditions
   when HotWordShow of this book is "Show"
      check menuItem "AlwaysShown"
      set sysHotwordsShown to true
   when HotWordShow of this book is "Hide"
```

```
      check menuItem "AlwaysHidden"
      set sysHotwordsShown to false
   else
      check menuItem "Show When Mouse Is In Window"
   end
end

to handle doHotWord what
   system FixRef, testing, s_suffixes, s_prefixes
   set sysCursor to 4
   set WrdTxt to what
   set oldPage to (this page)

   set wL to HotL(what)
   if wL is null then
      do
      if FailGo(what, "Def")
         if FailGo(what,"")
            set wLen to charCount(what)
            step i from 1 to textLineCount(s_suffixes)
                                          set hotW to what
                                          get textLine i of s_suffixes
               set sLen to charCount(it)
               if wLen > sLen+1 then
                        if chars wLen-sLen+1 to wLen of hotW is it
                           clear chars wLen-sLen+1 to wLen of hotW
                           if FailGo(hotW, "Def")
                              if FailGo(hotW,"")
                                 continue step
                              end
                           end
                           break doHotWord
                        end
                     end
            end

                                    -- Prefixes
            step i from 1 to textLineCount(s_prefixes)
                                          get textLine i of s_prefixes
                                          set hotW to it & what
               if FailGo(hotW, "Def")
                  if FailGo(hotW,"")
                     continue step
                  end
               end
               break doHotWord
         end
            break do
         end
      end
      break doHotWord
      until false

                  -- Failed to find hot word
      send failFind WrdTxt
      break doHotWord
```

```
    end if

    set Dest to null
    set p1 to offset("~@", wL)
    if p1 > 0 then
       if last character of wL is "@" then
          set Dest to characters 1 to p1-1 of wL
       else
          set Dest to characters p1 + 2 to charcount(wL) of wL
       end
    else
       set p1 to offset("~$", wL)
       if p1 > 0 then
         if last character of wl is "$" then
            set dest to characters 1 to p1-1 of wl
         else
            set dest to characters p1+2 to charcount(wl) of wl
         end
         send doPicture dest
         break doHotWord
       else
        set p1 to offset("~!", wL)
        if p1 > 0 then
          if last character of wl is "!" then
             set dest to characters 1 to p1-1 of wl
          else
             set dest to characters p1+2 to charcount(wl) of wl
          end
          send GoToPageWisdom dest
          break doHotWord
        else

          set p1 to offset("~", wL)
          if last character of wL is "~" then
             set Dest to characters 1 to p1-1 of wL
          else
             set Dest to characters p1 + 1 to charcount(wL) of wL
          end
          put space & "(Def.)" after Dest
        end
       end
    end

    if FailGo(Dest,"")
       send FailFind WrdTxt
    end if
end

to handle FailFind WrdTxt
    system testing, s_authorOK
    if testing
       send logBadWord WrdTxt to page 1
    else
       if s_authorOK
          send fix WrdTxt
       else
```

```
            request "Sorry, hot word not hooked up yet."
        end
    end
end

to handle SeekTopic what
    set failed to false
    set wL to HotL(what)
    if wL <> null then
        set Dest to characters (offset("~",wL) + 1) to 255 of wL
        set failed to FailGo(Dest, "")
    else
        if FailGo (what, "")
            set failed to FailGo(what, "Def")
        end
    end
    if failed
        request "Sorry, this topic is not available yet."
    end
end

to get SearchBackground bkgndName, startPage
    system SearchString

    set syslockscreen to true
    set syscursor to 4
    set sysHistoryRecord to false

    -- determine page range and get to first page
    set firstPage to first page of background bkgndName
    set lastPage to last page of background bkgndName
    if startPage is null or name of parent of startPage is not bkgndName
        set startPage to firstPage
    end
    if startPage is not this page
        go startPage
    end

    -- if text is selected (from a previous search) make sure we don't
    -- find something above it on the same page
    get selectedTextState
    if it <> null
        set p to item 2 of it + 1
    else
        set p to 0
    end
    clear focus

    -- starting with the current page search record fields for the
    -- search string
    set isFound to false
    do
        clear sysError
        search page records for searchString
        while selectedTextState <> null and sysError = null
            if item 2 of selectedTextState > p
```

```
            set isFound to true
            break do
         else
            search again
         end
      end

      set p to 0
      go next page of this background
   until this page is firstPage

   set sysHistoryRecord to true
   if this page is not first item of sysHistory
      push this page onto sysHistory
   end
   set sysCursor to default
   return isFound
end

to handle doPicture Dest

   set syscursor to 4
         -- see if wpicture is running
         getRemote "sysWindowHandle" application toolbook topic \
           "wpicture.tbk"
         set hWnd to it
         if it is null
      --find wbasemet.tbk
      clear sysError
      set syssuspend to false
      get book "wpicture.tbk"
      set syssuspend to true
      if syserror is not null
                  set syscursor to 1
         request  strNoPicture()
         break
      end

      --try running toolbook and the app
      set syssuspend to false
      clear syserror
      if sysruntime is true
          set exe to "tbook.exe"
         run "tbook.exe wpicture.tbk"
      else
          set exe to "toolbook.exe"
         run "toolbook.exe wpicture.tbk"
      end
          set syscursor to 1
      set syssuspend to true
      if syserror is not null
         request strExeNotInPath(exe)
         break
      end
      set syssuspend to false
      clear syserror
```

```
      keepRemote application "toolbook" topic "wpicture.tbk"
    else
      linkDLL user
        int ShowWindow(Word, int)
      end linkDLL
      get ShowWindow(hWnd,1)
    end if

    clear syserror
    executeRemote " send doImage"&&dest  application toolbook topic "wpicture.tbk"
    set syssuspend to true
    if item 1 of syserror is not "ok" then
      request item 1 of syserror
      break
    end if
    set syscursor to default
end

to handle goToPageWisdom Dest

    set syscursor to 4
          -- see if wpicture is running
          getRemote "sysWindowHandle" application toolbook topic \
            "wisdom.tbk"
          set hWnd to it
          if it is null
      --find wbasemet.tbk
      clear sysError
      set syssuspend to false
      get book "wisdom.tbk"
      set syssuspend to true
      if syserror is not null
                  set syscursor to 1
        request  strNowisdom()
        break
      end

      --try running toolbook and the app
      set syssuspend to false
      clear syserror
      if sysruntime is true
          set exe to "tbook.exe"
        run "tbook.exe wisdom.tbk"
      else
          set exe to "toolbook.exe"
        run "toolbook.exe wisdom.tbk"
      end
          set syscursor to 1
      set syssuspend to true
      if syserror is not null
        request strExeNotInPath(exe)
        break
      end
      set syssuspend to false
      clear syserror
      keepRemote application "toolbook" topic "wisdom.tbk"
```

```
    else
       linkDLL user
         int ShowWindow(Word, int)
         word setActiveWindow(Word)
       end linkDLL
       get ShowWindow(hWnd,1)
       get setactiveWindow(hWnd)
       unlinkDLL user
    end if

    clear syserror
    executeRemote " send goToPage"&&dest  application toolbook topic "wisdom.tbk"
    set syssuspend to true
    if item 1 of syserror is not "ok" then
       request item 1 of syserror
       break
    end if
    set syscursor to default
end
```

Script for Index Page

```
to handle buttonUp
   set L to item 1 of textFromPoint(argument 1)
   set P to 1 + item 2 of textFromPoint(argument 1)
   if L > 0 and P > 0 then
      set Lin to textline L of text of self
      set c to character p of Lin
      if c = "|" or c = " " then
         if P = 1 then
           set P to P + 1
         else
           set P to P - 1
         end if
      end if
      --set focus to self
      set c to character P of Lin
      set text of field "letter" of this background to c
      --search page for c
      if alphaIndex of this page <> null then
         set selectedTextLines of recordField "Topics" to\
          (item (chartoansi(c) - 64) of alphaIndex of this page)
      else
         if c = "A" then
            set focus to null
            search page records for "A"
         else
            do
               set focus to null
               set selection to null
               search page records for (LF & c)
               if selectedTextState <> null then
                  get recordfield "topics"
                  if charcount(text of it) - textoverflow of it \
                    < item 2 of selectedTextState
```

```
                increment scroll of it
            end
            break do
        else
            set c to ansiToChar(charToAnsi(c)-1)
        end if
      until c = "C"
    end if
  end if
end if
end buttonUp

to handle buttonUp
    --set c to textline selectedTextLines of my text
    set L to first item of textFromPoint(argument 1)
    if L > 0 then
        set c to textLine L of my text
    end
    if FailGo(c, "") and c contains ","
        -- inverted entry, maybe with lopped off article?
        set p to offset(",", c)
        set t1 to characters 1 to p - 1 of c
        set t2 to characters p + 2 to 1000 of c
        if FailGo (t2 && t1,"")
            if FailGo(t2 && "a" && t1, "")
                if FailGo(t2 && "an" && t1, "")
                    if FailGo(t2 && "the" && t1,"")
                        request "Sorry, can't find" && c
                    end
                end
            end
        end
    end
end buttonUp
```

Script for Picture Library

```
to handle EnterBook
  system  windowSized

  set syslockscreen to true
  set sysSuspend to true
  set syscursor to 4
  set sysLevel to reader
  hide scrollBar
    restore menuBar at both
    remove menu "file" at reader
    remove menu "edit" at reader
    remove menu "text" at reader
    remove menu "help" at reader
    remove menuitem "NewPage" at reader
    add menu "&File"  at reader
          add menuItem "&Compact Book" to menu "&File" at reader
    add menuItem "&Quit"  to menu "&File" at reader
  set sysFontFace to arial
  set sysFontSize to 10
  set sysFontStyle to bold
  set sysCursor to 1
  set sysDecimal to "."
  set sysDateFormat to "m/d/y"
  forward
  set sysChangesDB to false
  if windowSized is null then
    send sizeToPage
    set windowSized to true
  end if
  set syslockscreen to false
  set syscursor to default
end

to handle LeaveBook
    set pageDest of self to null
    restore system
end


to handle quit
--    send save
    send exit
end


to handle author
    if syslevel is "reader"
        set syslevel to author
        show menubar
        send sizetopage
        set sysChangesDB to true
        forward
```

```
    end
 end


 to handle reader
    if sysLevel is "Author"
       set sysLevel to reader
       send sizeToPage
       forward
    end
 end

 -- Handle F3 key at Reader level (Author not in menu)
 to handle keyDown x
    if x = keyF3
       send Author
    else
       forward
    end
 end



 to handle doImage dest
      set syscursor to 4
      set sysSuspend to false
      set sysError to null
      set it to dest
      if it is not null
        go page (characters 1 to 32 of it)
      end
      get sysError
      set sysSuspend to true
      set sysCursor to default
      if sysError is not null
        request syserror
        break to system
      end
      send showMyWindow
      set syscursor to default
 end

 to handle ShowMyWindow
    linkDLL user
       int IsIconic(word)
       int ShowWindow(Word, int)
       int SetActiveWindow(Word)
    end linkDLL
    if isIconic(sysWindowHandle) > 0 then
       get showWindow(sysWindowHandle, 1)
    end if
    get SetActiveWindow(sysWindowHandle)
    unlinkDLL user
 end
```

```
--squeeze the hell out of a modified book
--don't forget to change to the correct name of this book!
to handle compactBook
  get name of this book
  if it is null
    request "Can't compact a untitled book."
    break to system
  end if
  clear sysError
  save as "X.TBK",true
  if sysError is not null
     request "Save as X.TBK failed:"&&sysError
     break to system
  end if
  save as it ,true
  if sysError is not null
     request "Save as original name failed:"&&sysError
     break to system
  end if
end compactBook
```

Script for Page 1 of Picture Library

```
to handle buttonUp
   go next page
end buttonUp

to handle keyChar
   go next page
end keyChar

to handle enterpage
   set sysTimeFormat to "seconds"
   set my startT to sysTime
--   if syslevel is reader
--      show field "Credits"
--   end
end

to handle leavePage
         set sysLockScreen to true
--   hide field "Credits"
end

to handle idle
   -- don't go to main menu immediately, in case DDE command coming
   set sysTimeFormat to "seconds"
   if sysTime - my startT > 1 then
      go page "cover page"
   end
   forward
end idle
```

APPENDICES B

Instantaneous Center Calculation Algorithms

Instantaneous Center calculation for a Box shaped weld group

```
/*                                                    */
/* This program determines the ultimate resistance of a box shaped weld    */
/* group with a vertical load applied eccentrically in plane. A trial and */
/* error procedure using instantaneous centers and emperically determined  */
/* load deformation responses of weld elements is utilized.                */
/*                                                    */
/* The following variables are used in this program .....                  */
/*                                                    */
/*  l   = vertical length of box weld.                          */
/*  b   = horizontal width of box weld.                         */
/*  e   = eccentricity of load from c.g. of weld group.         */
/*  ro  = distance between i.c. and left side web of weld group.*/
/*  ri  = distance from i.c. to weld element.                   */
/*  rr  = distance to critical weld element (re: see variable delm)  */
/*  sum = sum of vertical reactions on weld group.              */
/*  sum1 = sum of moment reaction about i.c.                    */
/*  cc  = value of coefficient as used in handbook (table 3-27 , 1986)  */
/*  delm = maximum deflection of critical weld group element.   */
/*  deln = deflection of the n(th) weld group element.          */
/*  x1  = distance from i.c. to the right side web of weld group.*/
/*  thn = angle theta associated with the n(th) weld group element.  */
/*  rm  = ultimate strength of weld group element.              */
/*  rn  = resistance of weld group element.                     */
/*  mu  = regression coefficient.                               */
/*  lam = regression coefficient.                               */
/*  dpi = value of pi.                                          */
/*                                                    */

#include <math.h>
#include <stdio.h>
double calc(double , double , double );          /* function calc */
double sum, sum1;                          /* external variables */

main (int numb, char *geom[])
{
double l,b,e,cc,ro,rold,rnew,frold,frnew,secslp;
extern double sum, sum1;                   /* external variables */
FILE *out, *fopen();                       /* pointer to output file */

/*  convert input parameters (length, width, eccentricity) to float  */
l = atof(geom[1]);
b = atof(geom[2]);
e = atof(geom[3]);
printf("\n l: %f  b: %f  e: %f",l,b,e);
printf(" ");
printf("\n Program working ....  Please wait!");
```

```
/* scale weld to let l dimension equal to 200 mm.                    */
b=b*200.0/l;
e=e*200.0/l+b/2.0;
l=200.0;

ro = 0.0;            /* start with i.c. at left side web of weld group  */
sum = 0.0;                              /*  initialize counter  */
sum1 = 0.0;                             /*  initialize counter  */
calc(l,b,ro);                          /*  call to function calc  */
frold = (e+ro)*sum - sum1;             /* function of ro ; should be zero */
rnew = 10;                             /* place i.c. at new location */

while (fabs(frold) > 1000)  {              /** Continuously calculate **/
 rold = ro;                            /** new values of the fns. **/
 ro = rnew;                            /** frold and frnew until  **/
 calc(l,b,ro);                         /** the error is within    **/
 frnew = (e+ro)*sum - sum1;                  /** the specified amount.  **/
 secslp = (frnew - frold)/(rnew - rold);     /** The secant slope       **/
 rnew = rold - frold/secslp;                 /** (secslp) is used to    **/
 frold = frnew;                        /** get new values of ro.  **/
}

out = fopen("temp_c.out","w");
cc = sum1/(e+ro)*4.448*70/60/6.35/l*0.67*0.67;  /*  calculate coefficient  */
fprintf(out,"%lf",cc);
printf("%lf",cc);
fclose(out);                                /* close output file */
}

/* calc: function to test accuracy of guessed i.c. location          */
double calc(double l, double b, double ro)
{
 double delm,deln,x1,thn,rm,mu,lam,rn,dpi,r13,r2,rr,ri,delm1,delm2,delm3,i,j;
 extern double sum, sum1;                   /*  external variables  */

 dpi = 3.141592653589793;                   /*  specify value of pi  */
 x1 = ro + b;       /*  horizontal distance from i.c. to right side web  */

 /* check which weld element is critical: hor. weld (1) or ver. weld (2,3) */

 r2 = sqrt(pow(l/2.0,2)+pow(ro,2));         /*  radial distance from i.c. */
 r13 = sqrt(pow(l/2.0,2)+pow(x1,2));        /*  radial distance from i.c. */
 delm1 = 0.225*pow((atan2(2.0*x1,l)*180.0/dpi+5.0),-0.47);  /* max defln. */
 delm2 = 0.225*pow((atan2(l,2.0*ro)*180.0/dpi+5.0),-0.47);  /* max defln. */
 delm3 = 0.225*pow((atan2(l,2.0*x1)*180.0/dpi+5.0),-0.47);  /* max defln. */

 if (delm1/r13 < delm2/r2)  {        /* horizontal weld is critical .... */
  rr = r13;
  delm = delm1;
  }
 else  {                     /* left vertical weld is critical ... */
  rr = r2;
  delm = delm2;
        }
```

```
 if (delm3/r13 < delm/rr) {       /* right vertical weld is critical ... */
  rr = r13;
  delm = delm3;
  }

 sum = 0.0;                              /*  initialize counter  */
 sum1 = 0.0;                             /*  initialize counter  */

/*  sum of moments about i.c. and vertical forces for the horizontal welds  */

 for (i = ro+0.5 ; i <= x1+0.5 ; i++)
        {
        ri = sqrt(pow(l/2.0,2)+pow(i,2));     /* distance from i.c. to element */
        deln = ri/rr*delm;                    /*  deflection of n(th) element  */
        thn = fabs(atan2(2.0*i,l)*180.0/dpi);     /*  angle for n(th) element  */
        rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
        mu = 75.0*exp(0.0114*thn);              /*  regression coefficient  */
  lam = 0.4*exp(0.0146*thn);                /*  regression coefficient  */
  rn = rm*pow(1-exp(-mu*deln),lam);         /*  n(th) element strength  */
        sum = sum + 2.0*rn*i/ri;            /* add vertical force component */
        sum1 = sum1 + 2.0*rn*ri;                /*  add moment component  */
        }

/* sum of moments about i.c. and vertical forces for the left vertical weld */

 for (j = 0.5 ; j <= l/2+0.5 ; j++)
        {
        ri = sqrt(pow(ro,2)+pow(j,2));        /* distance from i.c. to element */
        deln = ri/rr*delm;                    /*  deflection of n(th) element  */
        thn = fabs(atan2(j,ro)*180.0/dpi);          /* angle for n(th) element */
  if (thn > 90.0)
          thn = 180.0 - thn;            /*  angle must be less than 90 degrees  */
        rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
        mu = 75.0*exp(0.0114*thn);              /*  regression coefficient  */
  lam = 0.4*exp(0.0146*thn);                /*  regression coefficient  */
        rn = rm*pow(1.0-exp(-mu*deln),lam);         /*  n(th) element strength  */
        sum = sum + 2.0*rn*ro/ri;           /* add vertical force component */
        sum1 = sum1 + 2.0*rn*ri;                /*  add moment component  */
        }

/* sum of moments about i.c. and vertical forces for the right vertical weld*/

 for (j = 0.5 ; j <= l/2+0.5 ; j++)
        {
  ri = sqrt(pow(x1,2)+pow(j,2));        /* distance from i.c. to element */
        deln = ri/rr*delm;                    /*  deflection of n(th) element  */
  thn = fabs(atan2(j,x1)*180.0/dpi);          /* angle for n(th) element */
  if (thn > 90.0)
          thn = 180.0 - thn;            /*  angle must be less than 90 degrees  */
        rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
        mu = 75.0*exp(0.0114*thn);              /*  regression coefficient  */
  lam = 0.4*exp(0.0146*thn);                /*  regression coefficient  */
        rn = rm*pow(1.0-exp(-mu*deln),lam);         /*  n(th) element strength  */
  sum = sum + 2.0*rn*x1/ri;             /* add vertical force component */
```

```
        sum1 = sum1 + 2.0*rn*ri;                /* add moment component */
        }

 return 0;              /* return 0 for the value of function calc */
}
```

Instantaneous Center calculation for a C_shaped weld group

```
/*                                                      */
/* This program determines the ultimate resistance of a C_shaped weld group */
/* with a vertical load applied in plane on the closed side. A trial and  */
/* error procedure using instantaneous centers and emperically determined  */
/* load deformation responses of weld elements is utilized.              */
/*                                                      */
/* The following variables are used in this program .....              */
/*                                                      */
/*  l   = length of web of weld group (ie: vertical length).          */
/*  b   = width of horizontal portion of weld group.                   */
/*  e   = eccentricity of load from web of weld group.                 */
/*  ro  = horizontal distance between i.c. and web of weld group.      */
/*  ri  = distance from i.c. to weld element.                          */
/*  rr  = distance to critical weld element (re: see variable delm)    */
/*  sum = sum of vertical reactions on weld group.                     */
/*  sum1 = sum of moment reaction about i.c.                           */
/*  cc  = value of coefficient as used in handbook (table 3-27 , 1986) */
/*  delm = maximum deflection of critical weld group element.          */
/*  deln = deflection of the n(th) weld group element.                 */
/*  x1  = horizontal distance from i.c. to weld group tips.            */
/*  thn = angle theta associated with the n(th) weld group element.    */
/*  rm  = ultimate strength of weld group element.                     */
/*  rn  = resistance of weld group element.                            */
/*  mu  = regression coefficient.                                      */
/*  lam = regression coefficient.                                      */
/*  dpi = value of pi.                                                 */
/*                                                      */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <owl.h>
#include "windows.h"
#include "wisdom.h"

long FAR PASCAL _export CweldCC (double l, double b, double e)
{
double cc,ro,rold,rnew,frold,frnew,secslp,up;
extern double sum, sum1;                /* external variables */


b=b*200.0/l;                            /* scale weld to let l = 200 */
e=e*200.0/l;
l=200;

ro = b;             /* start with i.c. in line with tips of weld group */
sum = 0.0;                              /* initialize counter */
```

```
sum1 = 0.0;                              /*  initialize counter  */
calcCloseC(l,b,ro);                          /*  call to function calc  */
frold = (e+ro)*sum - sum1;              /* function of ro ; should be zero */
rnew = b/2.0;                           /* place i.c. at new location */

while (fabs(frold) > 1000)  {               /** Continuously calculate **/
 rold = ro;                            /** new values of the fns. **/
 ro = rnew;                             /** frold and frnew until  **/
 calcCloseC(l,b,ro);                          /** the error is within    **/
 frnew = (e+ro)*sum - sum1;                /** the specified amount.  **/
 secslp = (frnew - frold)/(rnew - rold);      /** The secant slope       **/
 rnew = rold - frold/secslp;              /** (secslp) is used to    **/
 frold = frnew;                         /** get new values of ro.  **/
}


cc = sum1/(e+ro)*4.448*70/60/6.35/l*0.67*0.67;  /*  calculate coefficient  */
cc = cc * 1000;
up = ceil(cc);
return ((long) up);

}

/* calcCloseC: function to test accuracy of guessed i.c. location              */

double FAR PASCAL calcCloseC(double l, double b, double ro)
{
 double delm,deln,x1,thn,rm,mu,lam,rn,dpi,rr,ri,delm1,delm2,i,j;
 extern double sum, sum1;                        /*  external variables  */

 dpi = 3.141592653589793;                        /*  specify value of pi  */
 x1 = ro - b;       /*  horizontal distance from i.c. to weld group tips  */

 /* check which weld element is critical:  vert. weld (1) or hor. weld (2) */

 rr = sqrt(pow(l/2.0,2)+pow(ro,2));          /*  radial distance from i.c. */
 delm1 = 0.225*pow((atan2(l,2.0*ro)*180.0/dpi+5.0),-0.47);  /* max defln. */
 delm2 = 0.225*pow((atan2(2.0*ro,l)*180.0/dpi+5.0),-0.47);  /* max defln. */

 if (delm2 < delm1)     /* horizontal weld governs ultimate deflection.... */
        delm = delm2;
 else                  /* vertical weld governs ultimate deflection.... */
  delm = delm1;

 sum = 0.0;                                    /*  initialize counter  */
 sum1 = 0.0;                                    /*  initialize counter  */

/*  sum of moments about i.c. and vertical forces for the horizontal welds  */

 for (i = x1+0.5 ; i <= ro+0.5 ; i++)
        {
        ri = sqrt(pow(l/2.0,2)+pow(i,2));      /* distance from i.c. to element */
        deln = ri/rr*delm;                      /*  deflection of n(th) element  */
        thn = fabs(atan2(2.0*i,l)*180.0/dpi);      /*  angle for n(th) element  */
```

```
        rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /* ultimate strength */
        mu = 75.0*exp(0.0114*thn);                      /* regression coefficient */
  lam = 0.4*exp(0.0146*thn);                    /* regression coefficient */
  rn = rm*pow(1-exp(-mu*deln),lam);             /* n(th) element strength */
        sum = sum + 2.0*rn*i/ri;                /* add vertical force component */
        sum1 = sum1 + 2.0*rn*ri;                /* add moment component */
        }

/* sum of moments about i.c. and vertical forces for the vertical weld    */

 for (j = 0.5 ; j <= l/2+0.5 ; j++)
        {
        ri = sqrt(pow(ro,2)+pow(j,2));          /* distance from i.c. to element */
        deln = ri/rr*delm;                      /* deflection of n(th) element */
        thn = atan2(j,ro)*180.0/dpi;            /* angle for n(th) element */
        rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /* ultimate strength */
        mu = 75.0*exp(0.0114*thn);              /* regression coefficient */
  lam = 0.4*exp(0.0146*thn);                    /* regression coefficient */
        rn = rm*pow(1.0-exp(-mu*deln),lam);     /* n(th) element strength */
        sum = sum + 2.0*rn*ro/ri;               /* add vertical force component */
        sum1 = sum1 + 2.0*rn*ri;                /* add moment component */
        }

 return 0;              /* return 0 for the value of function calc */
}
```

Instantaneous Center calculation for a C_shaped weld group

```
/*                                                          */
/* This program determines the ultimate resistance of a C_shaped weld group */
/* with a vertical load applied in plane on the open side.  A trial and     */
/* error procedure using instantaneous centers and emperically determined   */
/* load deformation responses of weld elements is utilized.                 */
/*                                                          */
/* The following variables are used in this program .....                   */
/*                                                          */
/*  l   = length of web of weld group (ie: vertical length).               */
/*  b   = width of horizontal portion of weld group.                        */
/*  e   = eccentricity of load from web of weld group.                      */
/*  ro  = distance between i.c. and web of weld group.                      */
/*  ri  = distance from i.c. to weld element.                               */
/*  rr  = distance to critical weld element (re: see variable delm)         */
/*  sum = sum of vertical reactions on weld group.                          */
/*  sum1 = sum of moment reaction about i.c.                                */
/*  cc  = value of coefficient as used in handbook (table 3-27 , 1986)      */
/*  delm = maximum deflection of critical weld group element.               */
/*  deln = deflection of the n(th) weld group element.                      */
/*  x1  = horizontal distance from i.c. to the weld element tip.            */
/*  thn = angle theta associated with the n(th) weld group element.         */
/*  rm  = ultimate strength of weld group element.                          */
/*  rn  = resistance of weld group element.                                 */
/*  mu  = regression coefficient.                                           */
/*  lam = regression coefficient.                                           */
/*  dpi = value of pi.                                                      */
/*                                                          */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <owl.h>
#include "windows.h"
#include "wisdom.h"

long FAR PASCAL _export CweldOC (double l, double b, double e)
{
double cc,ro,rold,rnew,frold,frnew,secslp,up;
extern double sum, sum1;                    /*  external variables  */

b=b*200.0/l;                        /* scale weld to let l = 200 */
e=e*200.0/l;
l=200;

ro = 0.0;                   /* start with i.c. at web of weld group  */
sum = 0.0;                          /*  initialize counter  */
```

```
 sum1 = 0.0;                              /*  initialize counter  */
 calcOpenC(l,b,ro);                          /*  call to function calc  */
 frold = (e+ro)*sum - sum1;               /* function of ro ; should be zero */
 rnew = 10;                              /* place i.c. at new location */

 while (fabs(frold) > 1000)  {              /** Continuously calculate **/
  rold = ro;                             /** new values of the fns. **/
  ro = rnew;                              /** frold and frnew until  **/
  calcOpenC(l,b,ro);                          /** the error is within    **/
  frnew = (e+ro)*sum - sum1;                  /** the specified amount.  **/
  secslp = (frnew - frold)/(rnew - rold);      /** The secant slope       **/
  rnew = rold - frold/secslp;                /** (secslp) is used to    **/
  frold = frnew;                           /** get new values of ro.  **/
 }

 cc = sum1/(e+ro)*4.448*70/60/6.35/l*0.67*0.67;   /*  calculate coefficient  */
 cc = (cc*1000);
 up = ceil(cc);
 return ((long) up);

 }


 /* calcOpenC: function to test accuracy of guessed i.c. location              */

 double calcOpenC (double l, double b, double ro)
 {
  double delm,deln,x1,thn,rm,mu,lam,rn,dpi,r1,r2,rr,ri,delm1,delm2,i,j;
  extern double sum, sum1;                        /*  external variables  */

  dpi = 3.141592653589793;                        /*  specify value of pi  */
  x1 = ro + b;       /*  horizontal distance from i.c. to weld group tips  */

  /*  check which weld element is critical: vert. weld (1) or hor. weld (2) */

  r1 = sqrt(pow(l/2.0,2)+pow(ro,2));          /*  radial distance from i.c. */
  r2 = sqrt(pow(l/2.0,2)+pow(x1,2));          /*  radial distance from i.c. */
  delm1 = 0.225*pow((atan2(l,2.0*ro)*180.0/dpi+5.0),-0.47);   /* max defln. */
  delm2 = 0.225*pow((atan2(2.0*x1,l)*180.0/dpi+5.0),-0.47);   /* max defln. */

  if (delm2/r2 < delm1/r1)  {   /* horiz. weld governs ultimate deflection  */
   rr = r2;
   delm = delm2;
   }
  else  {                  /* vertical weld governs ultimate deflection  */
   rr = r1;
   delm = delm1;
         }

  sum = 0.0;                                    /*  initialize counter  */
  sum1 = 0.0;                                    /*  initialize counter  */

 /*  sum of moments about i.c. and vertical forces for the horizontal welds  */
```

```
  for (i = ro+0.5 ; i <= x1+0.5 ; i++)
          {
          ri = sqrt(pow(l/2.0,2)+pow(i,2));      /* distance from i.c. to element */
          deln = ri/rr*delm;                     /*  deflection of n(th) element  */
          thn = fabs(atan2(2.0*i,l)*180.0/dpi);      /*  angle for n(th) element  */
          rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;      /*  ultimate strength  */
          mu = 75.0*exp(0.0114*thn);                 /*  regression coefficient  */
   lam = 0.4*exp(0.0146*thn);                  /*  regression coefficient  */
   rn = rm*pow(1-exp(-mu*deln),lam);           /*  n(th) element strength  */
          sum = sum + 2.0*rn*i/ri;               /* add vertical force component */
          sum1 = sum1 + 2.0*rn*ri;                   /*  add moment component  */
          }

/*  sum of moments about i.c. and vertical forces for the vertical weld     */

  for (j = 0.5 ; j <= l/2+0.5 ; j++)
          {
          ri = sqrt(pow(ro,2)+pow(j,2));         /* distance from i.c. to element */
          deln = ri/rr*delm;                     /*  deflection of n(th) element  */
          thn = fabs(atan2(j,ro)*180.0/dpi);          /* angle for n(th) element */
   if (thn > 90.0)
            thn = 180.0 - thn;            /*  angle must be less than 90 degrees  */
          rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;      /*  ultimate strength  */
          mu = 75.0*exp(0.0114*thn);                 /*  regression coefficient  */
   lam = 0.4*exp(0.0146*thn);                  /*  regression coefficient  */
          rn = rm*pow(1.0-exp(-mu*deln),lam);         /*  n(th) element strength  */
          sum = sum + 2.0*rn*ro/ri;              /* add vertical force component */
          sum1 = sum1 + 2.0*rn*ri;                   /*  add moment component  */
          }

 return 0;                  /*  return 0 for the value of function calc  */
}
```

Instantaneous Center calculation for a L shaped weld group

```
/*                                                              */
/* This program determines the ultimate resistance of an L_shaped weld     */
/* group with a vertical load applied in plane on the closed side.  A trial */
/* and error procedure using instantaneous centers and emperically         */
/* determined load deformation responses of weld elements is utilized.     */
/*                                                              */
/* The following variables are used in this program .....              */
/*                                                              */
/*  l   = length of web of weld group (ie: vertical length).           */
/*  b   = width of horizontal portion of weld group.                    */
/*  e   = eccentricity of load from web of weld group.                  */
/*  ro  = distance between i.c. and web of weld group.                  */
/*  ri  = distance from i.c. to weld element.                   */
/*  rr  = distance to critical weld element (re: see variable delm)     */
/*  sum = sum of vertical reactions on weld group.                      */
/*  sum1 = sum of moment reaction about i.c.                            */
/*  cc  = value of coefficient as used in handbook (table 3-27 , 1986)  */
/*  delm = maximum deflection of critical weld group element.           */
/*  deln = deflection of the n(th) weld group element.                  */
/*  x1  = distance from i.c. to horizontal weld tip.                    */
/*  thn = angle theta associated with the n(th) weld group element.     */
/*  rm  = ultimate strength of weld group element.                      */
/*  rn  = resistance of weld group element.                             */
/*  mu  = regression coefficient.                                       */
/*  lam = regression coefficient.                                       */
/*  dpi = value of pi.                                          */
/*                                                          */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <owl.h>
#include "windows.h"
#include "wisdom.h"

long FAR PASCAL _export LweldCC (double l, double b,double e)
{
double cc,ro,rold,rnew,frold,frnew,secslp,up;
extern double sum, sum1;                    /*  external variables  */

b=b*200.0/l;                            /* scale weld to let l = 200 */
e=e*200.0/l;
l=200;

ro = b;          /* start with i.c. on the open side of the weld group  */
sum = 0.0;                              /*  initialize counter  */
```

```
sum1 = 0.0;                              /*  initialize counter  */
calcCloseL(l,b,ro);                          /*  call to function calc  */
frold = (e+ro)*sum - sum1;             /* function of ro ; should be zero */
rnew = b/2.0;                           /* place i.c. at new location */

while (fabs(frold) > 1000)  {              /** Continuously calculate **/
  rold = ro;                            /** new values of the fns. **/
  ro = rnew;                             /** frold and frnew until  **/
  calcCloseL(l,b,ro);                          /** the error is within    **/
  frnew = (e+ro)*sum - sum1;                /** the specified amount.  **/
  secslp = (frnew - frold)/(rnew - rold);      /** The secant slope       **/
  rnew = rold - frold/secslp;               /** (secslp) is used to    **/
  frold = frnew;                         /** get new values of ro.  **/
}

cc = sum1/(e+ro)*4.448*70/60/6.35/l*0.67*0.67;   /*  calculate coefficient  */
cc = (cc*1000);
up = ceil(cc);
return ((long)up);
}

/* calcCloseL: function to test accuracy of guessed i.c. location              */

double FAR PASCAL  calcCloseL(double l, double b, double ro)
{
 double x1,y1,thn,rm,mu,lam,rn,dpi,r1,r2,r3,rr,ri,i,j;
 double delm,delm1,delm2,delm3,deln;
 extern double sum, sum1;                      /*  external variables  */

 dpi = 3.141592653589793;                      /*  specify value of pi  */
 x1 = ro - b;       /*  horizontal distance from i.c. to weld group tips  */
 y1 = pow(l,2)/2/(l+b);

 /*  check which weld element is critical: vert. weld (1) or hor. weld (2) */

 r1 = sqrt(pow(l-y1,2)+pow(ro,2));          /*  radial distance from i.c. */
 r2 = sqrt(pow(y1,2)+pow(x1,2));            /*  radial distance from i.c. */
 r3 = sqrt(pow(y1,2)+pow(ro,2));            /*  radial distance from i.c. */
 delm1 = 0.225*pow((atan2(l-y1,ro)*180.0/dpi+5.0),-0.47);    /* max defln. */
 delm2 = 0.225*pow(fabs(atan2(x1,y1)*180.0/dpi+5.0),-0.47);  /* max defln. */
 delm3 = 0.225*pow((atan2(ro,y1)*180.0/dpi+5.0),-0.47);      /* max defln. */

 if (delm2/r2 < delm1/r1)  { /* horiz. weld (left side) governs deflection */
   rr = r2;
   delm = delm2;
   }
 else  {                  /* vertical weld governs ultimate deflection  */
   rr = r1;
   delm = delm1;
         }
 if (delm3/r3 < delm/rr)  {  /* hor. weld (right side) governs deflection  */
   rr = r3;
   delm = delm3;
   }
```

```
  sum = 0.0;                              /*  initialize counter  */
  sum1 = 0.0;                             /*  initialize counter  */

/*  sum of moments about i.c. and vertical forces for the horizontal welds  */

  for (i = x1+0.5 ; i <= ro+0.5 ; i++)
          {
   ri = sqrt(pow(y1,2)+pow(i,2));        /* distance from i.c. to element */
          deln = ri/rr*delm;                 /*  deflection of n(th) element  */
   thn = fabs(atan2(i,y1)*180.0/dpi);        /*  angle for n(th) element  */
          rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
          mu = 75.0*exp(0.0114*thn);              /*  regression coefficient  */
   lam = 0.4*exp(0.0146*thn);               /*  regression coefficient  */
   rn = rm*pow(1-exp(-mu*deln),lam);          /*  n(th) element strength  */
   sum = sum + rn*i/ri;                 /* add vertical force component */
   sum1 = sum1 + rn*ri;                     /*  add moment component  */
          }

/*  sum of moments about i.c. and vertical forces for the vertical weld     */

  for (j = y1-l+0.5 ; j <= y1+0.5 ; j++)
          {
          ri = sqrt(pow(ro,2)+pow(j,2));        /* distance from i.c. to element */
          deln = ri/rr*delm;                 /*  deflection of n(th) element  */
          thn = fabs(atan2(j,ro)*180.0/dpi);          /* angle for n(th) element */
   if (thn > 90.0)
            thn = 180.0 - thn;           /*  angle must be less than 90 degrees  */
          rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
          mu = 75.0*exp(0.0114*thn);              /*  regression coefficient  */
   lam = 0.4*exp(0.0146*thn);               /*  regression coefficient  */
          rn = rm*pow(1.0-exp(-mu*deln),lam);        /*  n(th) element strength  */
   sum = sum + rn*ro/ri;                /* add vertical force component */
   sum1 = sum1 + rn*ri;                     /*  add moment component  */
          }

  return 0;                /*  return 0 for the value of function calc  */
}
```

Instantaneous Center calculation for a L shaped weld group

```
/*                                                              */
/* This program determines the ultimate resistance of an L_shaped weld     */
/* group with a vertical load applied in plane on the open side.  A trial  */
/* and error procedure using instantaneous centers and emperically         */
/* determined load deformation responses of weld elements is utilized.     */
/*                                                              */
/* The following variables are used in this program .....                  */
/*                                                              */
/*  l   = length of web of weld group (ie: vertical length).               */
/*  b   = width of horizontal portion of weld group.                        */
/*  e   = eccentricity of load from web of weld group.                      */
/*  ro  = distance between i.c. and web of weld group.                      */
/*  ri  = distance from i.c. to weld element.                               */
/*  rr  = distance to critical weld element (re: see variable delm)         */
/*  sum  = sum of vertical reactions on weld group.                         */
/*  sum1 = sum of moment reaction about i.c.                                */
/*  cc  = value of coefficient as used in handbook (table 3-27 , 1986)      */
/*  delm = maximum deflection of critical weld group element.               */
/*  deln = deflection of the n(th) weld group element.                      */
/*  x1  = horizontal distance from i.c. to horizontal weld tip.             */
/*  thn  = angle theta associated with the n(th) weld group element.        */
/*  rm  = ultimate strength of weld group element.                          */
/*  rn  = resistance of weld group element.                                 */
/*  mu  = regression coefficient.                                           */
/*  lam  = regression coefficient.                                          */
/*  dpi  = value of pi.                                                     */
/*                                                              */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <owl.h>
#include "windows.h"
#include "wisdom.h"

long FAR PASCAL _export LweldOC (double l, double b, double e)
{
double cc,ro,rold,rnew,frold,frnew,secslp,up;
extern double sum, sum1;                    /*  external variables  */


b=b*200.0/l;                        /* scale weld to let l = 200 */
e=e*200.0/l;
l=200;
```

```
ro = 0.0;                    /* start with i.c. at web of weld group  */
sum = 0.0;                          /*  initialize counter  */
sum1 = 0.0;                         /*  initialize counter  */
calcOpenL(l,b,ro);                     /*  call to function calc  */
frold = (e+ro)*sum - sum1;          /* function of ro ; should be zero */
rnew = 10;                          /* place i.c. at new location */

while (fabs(frold) > 1000)  {              /** Continuously calculate **/
  rold = ro;                        /** new values of the fns. **/
  ro = rnew;                         /** frold and frnew until  **/
  calcOpenL(l,b,ro);                       /** the error is within    **/
  frnew = (e+ro)*sum - sum1;                /** the specified amount.  **/
  secslp = (frnew - frold)/(rnew - rold);      /** The secant slope      **/
  rnew = rold - frold/secslp;               /** (secslp) is used to    **/
  frold = frnew;                     /** get new values of ro.  **/
}

cc = sum1/(e+ro)*4.448*70/60/6.35/l*0.67*0.67;  /*  calculate coefficient  */
cc = (cc*1000);
up = ceil(cc);
return ((long) up);

}

/* calcOpenL: function to test accuracy of guessed i.c. location              */

double FAR PASCAL calcOpenL(double l, double b, double ro)
{
 double delm,deln,x1,y1,thn,rm,mu,lam,rn,dpi,r1,r2,rr,ri,delm1,delm2,i,j;
 extern double sum, sum1;                    /*  external variables  */

 dpi = 3.141592653589793;                    /*  specify value of pi  */
 x1 = ro + b;      /*  horizontal distance from i.c. to weld group tips  */
 y1 = pow(l,2)/2/(l+b);

 /*  check which weld element is critical: vert. weld (1) or hor. weld (2) */

 r1 = sqrt(pow(l-y1,2)+pow(ro,2));         /*  radial distance from i.c. */
 r2 = sqrt(pow(y1,2)+pow(x1,2));           /*  radial distance from i.c. */
 delm1 = 0.225*pow((atan2(l-y1,ro)*180.0/dpi+5.0),-0.47);   /* max defln. */
 delm2 = 0.225*pow((atan2(x1,y1)*180.0/dpi+5.0),-0.47);     /* max defln. */

 if (delm2/r2 < delm1/r1)  {   /* horiz. weld governs ultimate deflection  */
  rr = r2;
  delm = delm2;
  }
 else  {                /* vertical weld governs ultimate deflection  */
  rr = r1;
  delm = delm1;
        }

 sum = 0.0;                          /*  initialize counter  */
 sum1 = 0.0;                         /*  initialize counter  */
```

```
/*  sum of moments about i.c. and vertical forces for the horizontal welds  */

  for (i = ro+0.5 ; i <= x1+0.5 ; i++)
          {
    ri = sqrt(pow(y1,2)+pow(i,2));        /* distance from i.c. to element */
          deln = ri/rr*delm;                  /*  deflection of n(th) element  */
    thn = fabs(atan2(i,y1)*180.0/dpi);        /*  angle for n(th) element  */
          rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;      /*  ultimate strength  */
          mu = 75.0*exp(0.0114*thn);                /*  regression coefficient  */
    lam = 0.4*exp(0.0146*thn);                /*  regression coefficient  */
    rn = rm*pow(1-exp(-mu*deln),lam);          /*  n(th) element strength  */
    sum = sum + rn*i/ri;                  /* add vertical force component */
    sum1 = sum1 + rn*ri;                      /*  add moment component  */
          }

/*  sum of moments about i.c. and vertical forces for the vertical weld     */

  for (j = y1-l+0.5 ; j <= y1+0.5 ; j++)
          {
          ri = sqrt(pow(ro,2)+pow(j,2));        /* distance from i.c. to element */
          deln = ri/rr*delm;                  /*  deflection of n(th) element  */
          thn = fabs(atan2(j,ro)*180.0/dpi);          /* angle for n(th) element */
    if (thn > 90.0)
            thn = 180.0 - thn;           /*  angle must be less than 90 degrees  */
          rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;      /*  ultimate strength  */
          mu = 75.0*exp(0.0114*thn);                /*  regression coefficient  */
    lam = 0.4*exp(0.0146*thn);                /*  regression coefficient  */
          rn = rm*pow(1.0-exp(-mu*deln),lam);         /*  n(th) element strength  */
    sum = sum + rn*ro/ri;                 /* add vertical force component */
    sum1 = sum1 + rn*ri;                      /*  add moment component  */
          }

  return 0;                 /*  return 0 for the value of function calc  */
}
```

Instantaneous Center calculation for a 2 horizontal parallel weld group

```
/*                                                    */
/* This program determines the ultimate resistance of 2 horizontal parallel */
/* welds with a vertical load applied eccentrically in plane.  A trial and  */
/* error procedure using instantaneous centers and emperically determined   */
/* load deformation responses of weld elements is utilized.                 */
/*                                                    */
/* The following variables are used in this program .....                   */
/*                                                    */
/*  l   = length of horizontal weld line                                    */
/*  b   = distance between horizontal weld lines.                           */
/*  e   = eccentricity of load from c.g. of weld group.                     */
/*  ro  = distance between i.c. and left edge of weld group.                */
/*  ri  = distance from i.c. to weld element.                               */
/*  rr  = distance to critical weld element (re: see variable delm)         */
/*  sum = sum of vertical reactions on weld group.                          */
/*  sum1 = sum of moment reaction about i.c.                                */
/*  cc  = value of coefficient as used in handbook (table 3-25 , 1986)      */
/*  delm = maximum deflection of critical weld group element.               */
/*  deln = deflection of the n(th) weld group element.                      */
/*  xl  = distance from i.c. to the right edge of weld group.               */
/*  thn = angle theta associated with the n(th) weld group element.         */
/*  rm  = ultimate strength of weld group element.                          */
/*  rn  = resistance of weld group element.                                 */
/*  mu  = regression coefficient.                                           */
/*  lam = regression coefficient.                                           */
/*  dpi = value of pi.                                                      */
/*                                                    */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <owl.h>
#include "windows.h"
#include "wisdom.h"

long FAR PASCAL _export parHorC ( double l, double b , double e)
{
double cc,ro,rold,rnew,frold,frnew,secslp,up;
extern double sum, sum1;                    /*  external variables  */

b=b*200.0/l;                          /* scale weld to let l = 200 */
e=e*200.0/l+100.0;
l=200;

ro = 0.0;              /* start with i.c. at left side of weld group  */
sum = 0.0;                             /*  initialize counter  */
sum1 = 0.0;                            /*  initialize counter  */
```

```
calcParHor(l,b,ro);                          /*  call to function calc  */
frold = (e+ro)*sum - sum1;                  /* function of ro ; should be zero */
rnew = 10;                                  /* place i.c. at new location */

while (fabs(frold) > 1000)  {                /** Continuously calculate **/
 rold = ro;                                 /** new values of the fns. **/
 ro = rnew;                                 /** frold and frnew until  **/
 calcParHor(l,b,ro);                        /** the error is within    **/
 frnew = (e+ro)*sum - sum1;                 /** the specified amount.  **/
 secslp = (frnew - frold)/(rnew - rold);    /** The secant slope       **/
 rnew = rold - frold/secslp;                /** (secslp) is used to    **/
 frold = frnew;                             /** get new values of ro.  **/
}

cc = sum1/(e+ro)*4.448*70/60/6.35/l*0.67*0.67;  /*  calculate coefficient  */
cc = (cc*1000);
up = ceil(cc);
return ((long) up);
}

/* calcParHor: function to test accuracy of guessed i.c. location              */

double FAR PASCAL calcParHor(double l, double b, double ro)
{
 double delm,deln,x1,thn,rm,mu,lam,rn,dpi,rr,ri,i;
 extern double sum, sum1;                   /*  external variables  */

 dpi = 3.141592653589793;                   /*  specify value of pi  */
 x1 = ro + l;          /*  horizontal distance from i.c. to right side  */

 rr = sqrt(pow(b/2.0,2)+pow(x1,2));         /*  radial distance from i.c. */
 delm = 0.225*pow((atan2(2.0*x1,b)*180.0/dpi+5.0),-0.47);   /* max defln. */

 sum = 0.0;                                 /*  initialize counter  */
 sum1 = 0.0;                                /*  initialize counter  */

/*  sum of moments about i.c. and vertical forces for the horizontal welds  */

 for (i = ro+0.5 ; i < x1 ; i++)
        {
        ri = sqrt(pow(b/2.0,2)+pow(i,2));     /* distance from i.c. to element */
        deln = ri/rr*delm;                    /*  deflection of n(th) element  */
        thn = fabs(atan2(2.0*i,b)*180.0/dpi);     /*  angle for n(th) element  */
        rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
        mu = 75.0*exp(0.0114*thn);                /*  regression coefficient  */
  lam = 0.4*exp(0.0146*thn);                  /*  regression coefficient  */
  rn = rm*pow(1-exp(-mu*deln),lam);           /*  n(th) element strength  */
        sum = sum + 2.0*rn*i/ri;              /* add vertical force component */
        sum1 = sum1 + 2.0*rn*ri;                  /*  add moment component  */
        }

 return 0;                  /*  return 0 for the value of function calc  */
}
```

Instantaneous Center calculation for a 2 parallel vertical weld group

```
/*                                                    */
/* This program determines the ultimate resistance of two parallel vertical */
/* welds with a vertical load applied eccentrically in plane.  A trial and  */
/* error procedure using instantaneous centers and emperically determined   */
/* load deformation responses of weld elements is utilized.                 */
/*                                                    */
/* The following variables are used in this program .....                   */
/*                                                    */
/*  l   = length of vertical weld line.                                     */
/*  b   = horizontal spacing between weld lines.                            */
/*  e   = eccentricity of load from center of gravity.                      */
/*  ro  = distance between i.c. and left side web of weld group.            */
/*  ri  = distance from i.c. to weld element.                               */
/*  rr  = distance to critical weld element (re: see variable delm)         */
/*  sum = sum of vertical reactions on weld group.                          */
/*  sum1 = sum of moment reaction about i.c.                                */
/*  cc  = value of coefficient as used in handbook (table 3-27 , 1986)      */
/*  delm = maximum deflection of critical weld group element.               */
/*  deln = deflection of the n(th) weld group element.                      */
/*  xl  = horizontal distance from i.c. to right side of weld group.        */
/*  thn = angle theta associated with the n(th) weld group element.         */
/*  rm  = ultimate strength of weld group element.                          */
/*  rn  = resistance of weld group element.                                 */
/*  mu  = regression coefficient.                                           */
/*  lam = regression coefficient.                                           */
/*  dpi = value of pi.                                                      */
/*                                                    */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <owl.h>
#include "windows.h"
#include "wisdom.h"

long FAR PASCAL _export ParVerC ( double l, double b , double e)
{
double cc,ro,rold,rnew,frold,frnew,secslp,up;
extern double sum, sum1;                    /*  external variables  */

b=b*200.0/l;                          /* scale weld to let l = 200 */
e=e*200.0/l+b/2.0;
l=200;

ro = 0.0;             /* start with i.c. at left side web of weld group  */
sum = 0.0;                              /*  initialize counter  */
sum1 = 0.0;                             /*  initialize counter  */
```

```
calcParVer(l,b,ro);                          /*  call to function calc  */
frold = (e+ro)*sum - sum1;                 /* function of ro ; should be zero */
rnew = 10;                                 /* place i.c. at new location */

while (fabs(frold) > 1000)  {                /** Continuously calculate **/
 rold = ro;                                /** new values of the fns. **/
 ro = rnew;                                 /** frold and frnew until  **/
 calcParVer(l,b,ro);                              /** the error is within    **/
 frnew = (e+ro)*sum - sum1;                    /** the specified amount.  **/
 secslp = (frnew - frold)/(rnew - rold);       /** The secant slope       **/
 rnew = rold - frold/secslp;                 /** (secslp) is used to    **/
 frold = frnew;                             /** get new values of ro.  **/
}

cc = sum1/(e+ro)*4.448*70/60/6.35/l*0.67*0.67;   /*  calculate coefficient  */
cc = (cc*1000);
up = ceil(cc);
return ((long) up);
}

/* calcParVer: function to test accuracy of guessed i.c. location              */

double FAR PASCAL calcParVer(double l, double b, double ro)
{
 double delm,deln,x1,thn,rm,mu,lam,rn,dpi,rr,ri,j;
 extern double sum, sum1;                        /*  external variables  */

 dpi = 3.141592653589793;                        /*  specify value of pi  */
 x1 = ro + b;       /*  horizontal distance from i.c. to right side web  */

 /* check which weld element is critical: hor. weld (1) or ver. weld (2,3) */

 rr = sqrt(pow(l/2.0,2)+pow(x1,2));          /*  radial distance from i.c. */
 delm = 0.225*pow((atan2(l,2.0*x1)*180.0/dpi+5.0),-0.47);   /* max defln. */

 sum = 0.0;                                  /*  initialize counter  */
 sum1 = 0.0;                                 /*  initialize counter  */

/* sum of moments about i.c. and vertical forces for the left vertical weld */

 for (j = 0.5 ; j < l/2 ; j++)
        {
        ri = sqrt(pow(ro,2)+pow(j,2));        /* distance from i.c. to element */
        deln = ri/rr*delm;                   /*  deflection of n(th) element  */
        thn = fabs(atan2(j,ro)*180.0/dpi);          /* angle for n(th) element */
  if (thn > 90.0)
         thn = 180.0 - thn;           /*  angle must be less than 90 degrees  */
        rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
        mu = 75.0*exp(0.0114*thn);                /*  regression coefficient  */
  lam = 0.4*exp(0.0146*thn);                 /*  regression coefficient  */
        rn = rm*pow(1.0-exp(-mu*deln),lam);        /*  n(th) element strength  */
        sum = sum + 2.0*rn*ro/ri;             /* add vertical force component */
        sum1 = sum1 + 2.0*rn*ri;                  /*  add moment component  */
        }
```

```
/* sum of moments about i.c. and vertical forces for the right vertical weld*/

  for (j = 0.5 ; j <= l/2+0.5 ; j++)
          {
    ri = sqrt(pow(x1,2)+pow(j,2));        /* distance from i.c. to element */
          deln = ri/rr*delm;                  /*  deflection of n(th) element  */
    thn = fabs(atan2(j,x1)*180.0/dpi);          /* angle for n(th) element */
          rm = (10.0 + thn)/(0.92 + 0.0603*thn)/25.4;     /*  ultimate strength  */
          mu = 75.0*exp(0.0114*thn);                /*  regression coefficient  */
    lam = 0.4*exp(0.0146*thn);                /*  regression coefficient  */
          rn = rm*pow(1.0-exp(-mu*deln),lam);        /*  n(th) element strength  */
    sum = sum + 2.0*rn*x1/ri;              /* add vertical force component */
          sum1 = sum1 + 2.0*rn*ri;                  /*  add moment component  */
          }

  return 0;                 /*  return 0 for the value of function calc  */
}
```