AIMMS User's Guide - Designing End-User Interfaces

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit www.aimms.com or order your hard-copy at www.lulu.com/aimms.

Paragon Decision Technology B.V.
Schipholweg 1
2034 LS Haarlem
The Netherlands
Tel.: +31 23 5511512
Fax: +31 23 5511517

Paragon Decision Technology Inc.
500 108th Avenue NE
Ste. # 1085
Bellevue, WA 98004
USA
Tel.: +1 425 458 4024
Fax: +1 425 458 4025

Paragon Decision Technology Pte. Ltd.
80 Raffles Place
UOB Plaza 1, Level 36-01
Singapore 048624
Tel.: +65 9640 4182

Email: info@aimms.com
WWW: www.aimms.com

This documentation was typeset by Paragon Decision Technology B.V. using LATEX and the LUCIDA font family.

Chapter 15

Designing End-User Interfaces

This chapter

The goal of this chapter is to give some directions that will help you design an end-user interface that is both easy to use and straightforward to maintain. Even though design is an intuitive subject, you may find that developing a professional interface is no trivial matter. The general design principles outlined in this chapter, as well as the tools provided by AIMMS (e.g. the **Template Manager**, the **Page Manager** and the page design tools), will help you to specify and maintain a high-quality model-based interactive report.

Linear design is not recommended

A linear design process consists of the following steps.

- Develop an extensive technical specification of the interface following consultation with prospective users.
- Let the prospective users read and evaluate the specification.
- Adjust the specification and implement the interface accordingly.

Even though this approach may seem natural, in practice it does not work well. Users are usually unable to specify precisely what they require, and they find it difficult to imagine the look and feel of a graphical user interface from a written document. That is why we recommend an iterative design process based on prototyping.

Design through prototyping recommended

Rather than writing a detailed technical specification of a graphical user interface you can construct an initial design immediately following a consultation with prospective users. You will find that the evaluation of a well-designed prototype is a much better way of helping users structure their wishes, and it provides them with an impulse for new ideas. With AIMMS' point- and-click tools the actual construction of, and subsequent adjustments to, your interface are not a major task. As a result, you will be able to complete an interface in a limited number of iterations with the assurance that it will be accepted by your final users.

Main subjects

This chapter focuses on three major topics related to interface design, each of which is covered in a separate section. These are

- page design,
- navigation between pages, and

- quality of interaction with your end-user.

In the first section of this chapter these three topics are briefly introduced with a quick orientation and summary. In the final section you will find some pages taken from real-life applications in order to illustrate the principles discussed in this chapter.

15.1 Quick orientation

This section

This section gives you a quick and pragmatic overview of some of the considerations you should take into account when designing an end-user interface.

Page design for yourself

If you design an interface for yourself, your only concern is to improve your own efficiency while developing and debugging your AIMMS model. In this case the following guidelines are relevant.

- Design one or more single-page overviews combining related input and output data on the same page.
- Use a template page to setup and link the different data pages.
- Link procedures that perform the various data manipulations required by your model using buttons on the relevant pages.

Page design for someone else

Should you be designing the interface for someone else, then your basic concern is to make sure that your design will be used effectively. The key factor here is the quality of communication. The interface should possess the following characteristics:

- a pleasant look and feel (all pages use the same layout and all objects have clear descriptions in their title and status line),
- a consistent navigational structure (making the behavior of the system predictable), and
- robustness (through clear error messages and extensive error checking).

These characteristics should lead to user-friendliness and thereby to acceptance of the interface by your end-users.

Navigation with few pages

The number of pages in an application will depend upon the size and complexity of the model, as well as on the number of options you want to provide to your end-users. In an application with few pages the navigation structure can be quite simple. For example, a wheel-like structure linking all pages will suffice. In such a structure each page is linked to the previous, the next, and the main page, with the main page providing direct access to all other pages.

Navigation with many pages

If the application consists of a large number of pages, then a wheel-like structure with a single main page is not practical. In this case, a tree structure with cross links between the pages is a good option to facilitate ease of navigation. The following characteristics contribute directly to the success of such a structure:

- main sections that are easily accessible and subsections that contain their own outlines or menus,
- section headings in a fixed position on every page,
- a customized menu bar that can be used to give quick access to important pages as well as to provide menu items for general actions that can be executed from any page (for instance a print command), and
- extra orientation clues by associating colors (for instance in the title bar) with the different sections.

Interaction with occasional users

Occasional users will not require much control over the behavior of the model. They view the interface as an easy way of browsing through information, and occasionally carrying out some experiments. An appropriate interface should encompass the following characteristics:

- easy to read with clear explanations attached to symbols and icons,
- summaries of important model results,
- graphs for quick trend perception,
- no advanced control options requiring explanation, and
- a tree structure of pages enabling occasional users to quickly find information.

Interaction with frequent users

Frequent users usually know a lot about the underlying application. They tend to use the application as an operational tool, and are prepared to spend some time learning how to use it. This will allow you to build more advanced functionality into the interface. In addition to those characteristics mentioned above, we suggest that the following characteristics should be included:

- advanced control options for quick access throughout the interface,
- page design adjusted to familiar report formats,
- use of existing color conventions,
- a wide tree structure with relatively few levels,
- a setup to simplify the import of new data, and
- a help system, or a number of separate pages displaying help text.

What's next

In the next three sections the subjects of *page design*, *navigation*, and *interaction* will be discussed in more detail. These sections, together with the summary provided in this section, form a basis on which you should be able to design a high-quality interface for your end-users.

15.2 Page design

General page design principles

First, we will provide a few general guidelines for page design. Some of them will be elaborated in the subsequent paragraphs.

- All pages should use the same layout. Important buttons appearing on every page should always appear in the same place. The easiest way of achieving this is to use page templates.
- Give pages a clear title. The title is often the user's first clue as to the contents of the page.
- In order to clarify the meaning of each page place comments in the status line, title or descriptive text of each page object. Additionally, do not put too much text on a page.
- Limit the use of colors for titles and areas to a balanced combination of two to four colors.
- Use only those fonts that are installed on all computers, and pay attention to their readability.

Divide page into areas

The layout of pages throughout an interactive report will become more consistent if you divide each page into different areas each containing a group of related objects. There are several areas that you could consider:

- the page title and section indicator,
- the data object(s),
- navigational buttons (previous, next, main, go back),
- buttons with actions such as checks and calculations,
- a logo,
- one or more floating indices, and
- a reference to a certain model state, currently loaded data set, etc.

Visually, you can use borders and/or colors to highlight these areas. Two typical examples of how to divide a page are given in Figure 15.1.

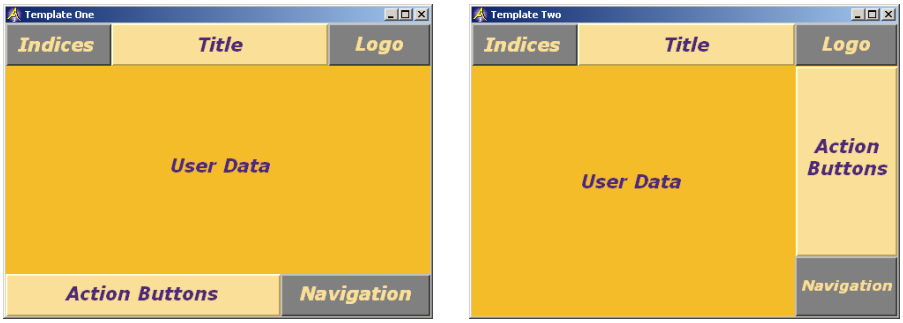

Figure 15.1: Example page layouts

Bitmap buttons

By using bitmaps on buttons users can quickly recognize their meaning. For example, most people will interpret an arrow on a button faster than the word 'Next'. Sometimes the combination of an icon and a text is preferred. Initially, users will tend to read the text to identify the button's action, while later on just the icon will suffice. You should ensure that both the style and the size of the bitmap used in your interface are consistent, and that you do not clutter the interface with too many different bitmaps.

Color is important

Experience has shown that the use of color is crucial in the acceptance of your interface. The golden rule is to be sensitive to the wishes of the end-users, and to use combinations of colors that most people will appreciate (not an easy task). Suggestions on your part could reflect such items as house style, logo, or colors that everyone automatically associates with the underlying application.

Some guidelines

Some guidelines concerning the use of color on a single page are as follows.

- Colors can be used as a way to visually segment the page into separate regions, or to draw the user's attention to a particular point.
- Even though you may be tempted to use lots of different colors, it is wise not to do so. Too many colors will clutter the screen and tire the eyes.
- The contrast between foreground and background colors must be sufficient to facilitate reading.
- Selected colors should not conflict with familiar interpretations.
- Color choice should always be such that color-blind users are able to use the interface without problem.

Decide on a coloring palette

A predetermined coloring palette will give each page a consistent look. You will need colors for *page format*, *page text* , and *page highlights*.

- Format colors make up the page backgrounds. They are colored rectangles behind data objects, text and logos. Light gray is frequently used as a background color, because buttons look good and shadow effects come up nicely.
- Text colors are used for titles, foreground color in data objects, etc. There should be a clear contrast with the format colors.
- Highlight colors are used for the remaining objects. In AIMMS you can specify your own color schemes and link these to particular data objects.

User-adjustable colors

If your application is used by many users, it may be impossible to satisfy the color preferences of all users. In that case, you can define all colors in the interface through color parameters, and create a color setup page in which the user can select his preferred color scheme.

Fonts are also important

The size and style of the fonts will directly affect the look and readability of each page. Just as with colors you should avoid using too many different fonts, as they will give the interface a disorganized look.

Some guidelines

Here are some extra points you may wish to consider when selecting fonts for objects on a page.

- Generally, sans serif fonts (such as Arial) are more readable on a computer screen than fonts with serifs (i.e. fonts with small wedge-shaped extensions to each character such as Times). This is particularly true for small font sizes.
- Similarly, regular fonts are more readable on a computer screen than italic fonts.
- Words in (the familiar) lower case are easier to read than words in capitals. Words in upper case can be used to attract attention as long as they are used sparingly.
- Vary the size of characters for emphasis, but try to limit the number of sizes. In this way a user will recognize the implied hierarchical structure behind the text.

Give fonts functional names

AIMMS allows you to put together your own font list, and give names to the fonts. You can take advantage of this facility by naming fonts in terms of their functionality. Typical examples are fonts named “page title”, “button” and “table”. This will help you to make consistent font choices for each object during page construction. Should you subsequently decide to change the fonts during the maintenance phase of your interface, then all you need to do is to edit the font in the font list, and all objects with that font name will be automatically updated.

Be aware of technical limitations

The number of colors supported by the video display adaptor determines the possibilities in using colors. The screen resolution of the monitor determines the size and contents of pages. Not all available fonts are installed on every computer. It is wise to take these technical limitations into account by checking the hardware limitations of your end-users. You could always use fewer shades of color, design for lower resolution, and limit your choice to standard fonts.

15.3 Navigation

General navigation principles

An AIMMS application basically consists of pages that are linked by buttons. These pages should be presented in an order that is both meaningful and logical to the end-user. This is where navigation becomes important. The AIMMS **Page Manager** helps you with navigation. The following general guidelines may be helpful.

- The performance of the system should be predictable. A user will create a mental picture based on his experience with the current, and similar, systems. Try to adhere to standards set by other systems.
- A user should always be able to return to the page he just left. AIMMS offers a specific button action for this purpose.
- Give the user easy access to page overviews from buttons placed on every page, or via submenus (accessible from every page).
- When the number of pages is small, use a wheel structure to navigate. All pages are then linked through buttons to the previous and the next page, as well as to a single main page from which all other pages are accessible.
- When the number of pages is large, use a tree structure to navigate. Then the number of steps needed to arrive at any particular page is at most the number of levels in the tree. The wheel structure can still be used for small self-contained subsets of pages.

Navigate between sections

When linking pages to improve navigation throughout the interface, it helps to distinguish sections of pages that belong together. Typical sections are:

- *input sections* enabling an end-user to view, edit and enter data,
- *output and experimentation sections* to present model results, derived data, summaries and results pertaining to multiple cases, and
- *control sections* for guiding the model configuration, the execution of model runs, and the printing of customized reports.

15.4 Interaction

This section

This section gives you some further design guidelines which will have a positive impact on the quality of interaction with end-users.

Know your users

One of the most important principles in user interface design is to know your users. When you consider that the interface is an intermediary between your model and the end-user, you will realize that it is a means of communication. Therefore it is essential that you carefully

- identify the needs of your users,
- study their standards of communication,
- consider their level of knowledge of the application area, and
- recognize their abilities with their computer.

The more you can accommodate your end-users' needs, the more it will reduce their learning time and improve their acceptance of the system.

What to emphasize

Once you know your users, you will know how to address them in the interface. Several relevant aspects are:

- the symbols or text used on buttons to indicate their actions,
- the amount of guidance in the form of message dialog boxes,
- the existence of fixed sequences to carry out certain tasks,
- the existence and style of feedback messages, and
- the use of existing color conventions or symbols for certain products or status parameters.

Start-up procedure

The initial interaction with your end-users in an interface should occur without any knowledge on their part. That is why you should create a start-up procedure that runs automatically on opening the project. Typical actions that may be included in such a procedure are:

- importing relevant data,
- executing required initial calculations,
- opening the correct first page, and
- setting up the first dialog box.

Interaction should be robust

Users can become frustrated and discouraged if they work with a system in which solutions can become infeasible, input errors are not detected, or results somehow get lost. You could improve your interaction with the user by applying the following guidelines.

- Declare upper and lower bounds for parameters and variables. When your users enter values outside these bounds, AIMMS will automatically produce an error message.
- Write error checks in procedures. These procedures can be called after a user updates a data object. If an error is detected, the procedure can issue a message.
- Provide clear and explicit diagnostics once an error is detected.
- If your end-users are not allowed to modify particular parameter values, make these parameters read-only in the interface.
- Avoid the possibility that models become infeasible by introducing extra "slack" or "surplus" variables. In addition, provide on-screen messages when these variables become positive.
- Always ask for confirmation if a user attempts to carry out an irreversible action.
- Use message dialog boxes to motivate your users to save a case after solving the model, so that they can always revert to a saved case.

Select the right object

For each identifier on a page you must select the appropriate object for its display. The following hints may be helpful.

- Use tables or scalar objects when it is important to display exact figures, or when the object will often be used for data entry.
- Use composite table objects for identifiers with many dimensions and few nonzeros. You can also use them for the display of multidimensional (sub)sets.
- Use pivot table objects if you want to view or compare identifiers with a common index domain, or if you want an object that is highly customizable by the end-user.
- Use bar charts, curves and Gantt charts for compact overviews of model results. Both curves and Gantt charts are ideal for presenting time-dependent data. Bar charts are appropriate for displaying the relative size of items. In addition to charts consider supplying extra pages with the same information in tabular form, so that exact values can be read and modified.
- Use Gantt charts when you want to combine a lot of information in one chart, or when you want to display ordering information in a sequence.
- Use network flow objects to provide a visual overview of results in any application in which flows between objects play a role.
- Use a stacked bar chart to show how a series of components add up to form a whole.
- Use a selection object or a table with 0/1 values for yes/no decisions. Both have the advantage that you can change values by single or double mouse clicks.
- Use selection objects for all situations where it is more meaningful for a user to select a description from a list rather than entering a number. If you do this, you may have to declare some extra sets, or parameters, for display purposes.
- Use AIMMS' capabilities of linking indices to element parameters to show multidimensional data. This gives you the opportunity of displaying large amounts of data in a concise way by using identifier slices fixed to one or more element parameters, and showing the data for the remaining indices only.

In all cases, it is important that you structure the information within an object in a meaningful manner. You should make deliberate decisions regarding the selection of row and column labels in a table, the choice of the x-axis in a bar chart, the number and the display of grid lines, the benefit of removing zeros, etc.

15.5 Illustrative example pages

A flow chart page

In Figure 15.2 you find an example of a page displaying a schematic representation of the scope of the model. This not only provides the user with an insight into the process being modeled, it also serves as a menu page. The user can click on a tank or an arrow to jump to the corresponding section of pages.

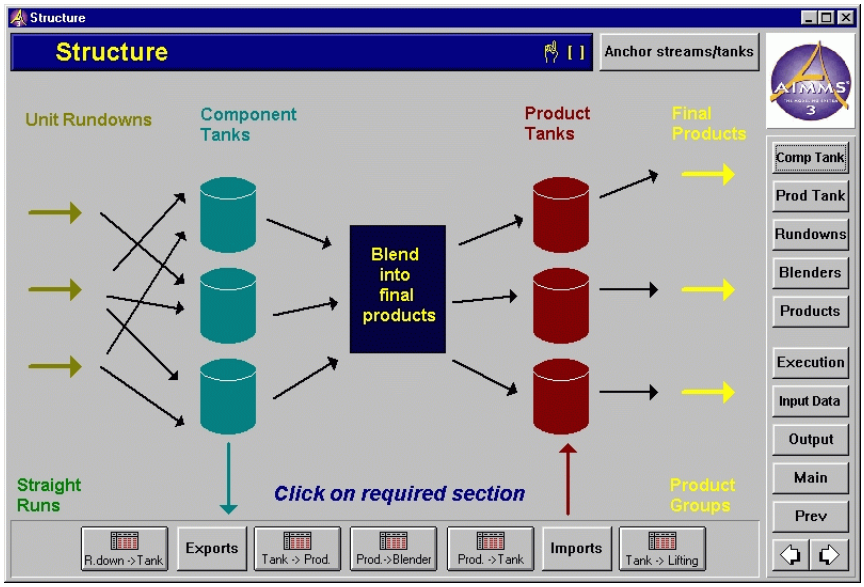

Figure 15.2: A flow chart page

A Gantt chart page

Figure 15.3 is an example of a page displaying tasks scheduled by the model in the form of two related Gantt charts. The example page is taken from a highly interactive application that is used to quickly propagate the consequences of manual changes that are made by schedulers.

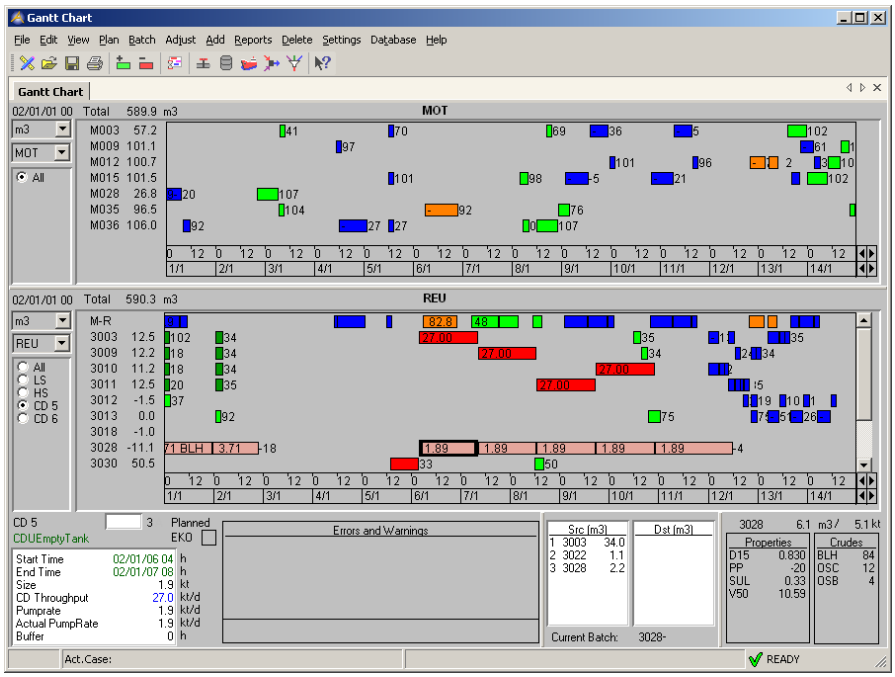

Figure 15.3: A Gantt chart page

The area on the left of each Gantt chart contains some controls that allows the user to change what is being displayed: one or more radio buttons that select the object for which tasks are being displayed and a drop-down list that allows the user to change the display unit of the numbers that are being displayed. The x-axis of the Gantt chart is a time axis representing both hours and days. By clicking on one of the batches in either Gantt chart, detailed information about the task is displayed in the lower part of the window. Note that the page is equipped with a user menu (and toolbar) that allows the user to perform all kinds of batch related operations.

A list page

On the page shown in Figure 15.4 is a list object that is used to display a list of values. The first two columns are associated with different cases, the third column displays the difference between the two. The units associated with the identifiers are shown to the right. In this page button areas are positioned along the bottom and right side of the page.

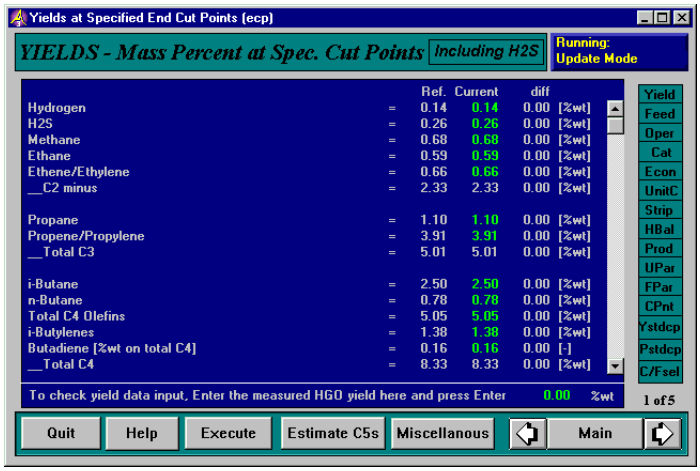

Figure 15.4: A page with lists of values and buttons

A page controlling a report

The page in Figure 15.5 contains a number of selection objects. With these objects an end-user can indicate which sections should be included in a printed report. The file name object in the top right corner displays the report's filename. The user can change this name after clicking on the icon.

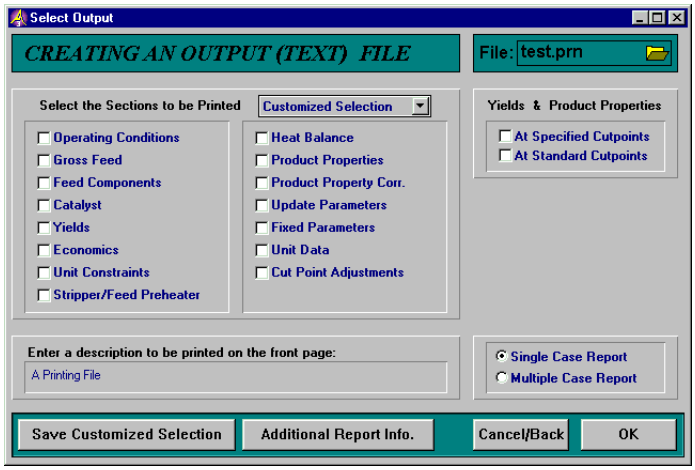

Figure 15.5: A page controlling the sections to be printed in a report

A page visualizing a production flow

Figure 15.6 shows a page from a production planning application for a steel industry reduction area (including a sintering plant and blast furnaces). The application involves the integrated planning of raw material preparation, sintering, and operations at blast furnaces. An optimization model, using a customized heuristic that sequentially solves basic MILP (mixed integer linear programming) and NLP (nonlinear programming) sub-problems, determines the quantities of raw material inputs for the production of sinter and pig iron. Nonlinear constraints are included in the model in order to compute the fuel mix for the sintering plant and blast furnaces, as well as the distribution of sinter among the blast furnaces. Integer variables were included in the model in order to represent whole train cargoes of mineral ore and operational alternatives of coal mixtures for pulverized injection.

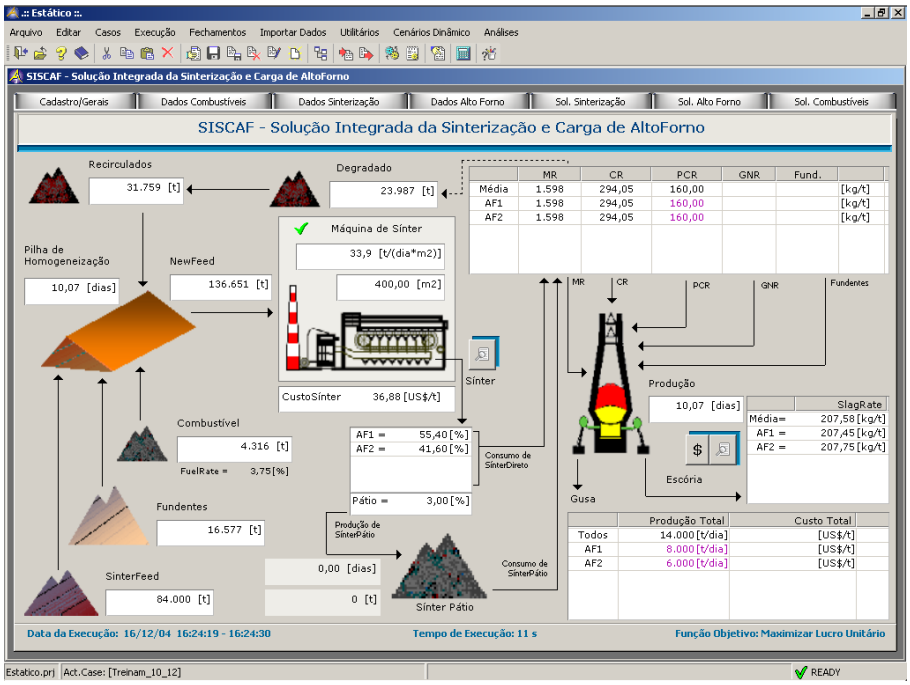

Figure 15.6: A page with several data objects, buttons and images

A collection of tables and scalar object combined with illustrative images visualizes a recommended plan for the production of sinter and pig iron out of raw materials. The seven text buttons on top of the page title can be used to navigate through the application. Note that the interface has been developed using a Brazilian character set.