Designing the Application Structure

The design of dialog functions is one of the most important activities at the beginning of a design phase. It is separated into the following tasks:

- Definition of the application structure.
- Separation of the system into business functions.
- Definition of the structure of the individually callable components.
- Decision on which frames will be used for which business functions.

The following topics are covered below:

- Familiarizing Yourself with the Application Frames
- The Business Function

Familiarizing Yourself with the Application Frames

Before designing the application structure, you should make sure you are familiar with the application frames. The selected frame determines the basic functionality of a function or a processing step. Frame selection is therefore a very important step in the transition between the functional design phase and implementation.

In order to select frames, you need:

- an overall understanding of which frames are available;
- an understanding of the possible combinations of program frames which can be used when; creating a function.
- to identify reusable components (this reduces the effort involved in frame selection);
- to consider the specific functional requirements for each function.

The Business Function

A business function is the smallest individually callable application unit. It can be an administrative function for maintaining data, an action such as making a reservation, or an enquiry. Depending on requirements, it could consist of a single dialog or a main dialog and a number of subordinate dialogs.

Each dialog can be supported by additional functionality such as background processing or a sequence of further lower level windows (e.g. selection help).

- Identifying a Business Function
- Structuring a Function

Identifying a Business Function

A business function acts on a group of related data, subsequently referred to as an object. An object can contain fields from various physical data sources.

A business function can include several actions for an object (for example, modify or copy), or be limited to one action (for example, delete).

Structuring a Function

Functions can be broken down into individual dialogs. You select the frame for each dialog to be used depending on the functionality required.

You should consider the following guidelines when creating functions:

- Use as few dialogs as possible.
- Place logically related fields on the same window.
- Place mandatory fields on the maintain dialog whenever possible. If there is insufficient space, or if no logical structure is evident, then place the mandatory fields within a subordinate dialog which you indicate as important. For example, via a push button.
- Determine if a function contains lower level dialogs that are shareable with other functions. Define such common functionality as separate reusable dialogs.
- Make such reusable modules available to the project team as soon as possible. Use subprograms for related business functionality which is more of a business nature. Use subordinate dialogs or modal windows for related business functionality which must be invoked by the end user as a dialog.
- Check for functionality which can be optionally invoked by the end user, and which are not required for the normal completion of the function. Define such functionality as subordinate dialogs or as modal windows.
- Check the necessity of selection help for the individual fields. Selection help for key fields should always be available.
- Check if confirmation or information windows are to be used.
- Determine the necessary frame functionality which is required.

- Build the call mechanisms for lower level dialogs in a way which is logical when seen from the business perspective (example, push buttons in prioritized sequence).

