

TABLE MODIFIABLE EDIT FUNCTIONS WITH ORDER-EFFECTIVE EDIT RULES

This application is a continuation of prior application Ser. No. 07/443,617, filed on Nov. 30, 1989, now abandoned.

The present invention is related to the following inventions, all assigned to the assignee of the present invention:

System Control Structure of a Hospital Information System and Method of Using Same, having Ser. No. 116,614, and filed on Nov. 3, 1987, now abandoned.

Method for Generating Patient-Specific Flowsheets by Adding/Deleting Parameters, having Ser. No. 116,611, and filed on Nov. 3, 1987, now U.S. Pat. No. 4,878,175;

Clinical Task List with Charting Through the Task List onto Underlying Form and Automatic Updating of Task List, having Ser. No. 268,822, and filed on Nov. 7, 1988, now U.S. Pat. No. 5,077,666.

Clinical Task List with Charting onto Underlying Form and Automatic Updating of Task List, having Ser. No. 268,323, and filed on Nov. 7, 1988, now U.S. Pat. No. 5,072,383;

Method For Generating A Display, having Ser. No. 322,740, and filed on Mar. 13, 1989, now abandoned, and continuation Ser. No. 07/540,382, now U.S. Pat. No. 5,208,907;

Electronic Data Storage Interface, having Ser. No. 408,178 and filed on Sep. 15, 1989, now U.S. Pat. No. 5,208,907;

Method for Updating Data in a Database, having Ser. No. 07/408,167 and filed on Sep. 15, 1989, now abandoned;

A Method for Displaying Information from an Information Based Computer System, having Ser. No. 407,979, and filed on Sep. 15, 1989, now abandoned;

Spreadsheet Cell having Multiple Data Fields, having Ser. No. 408,166, and filed on Sep. 15, 1989, now abandoned;

A Method for Displaying Information from an Information Based Computer System, having Ser. No. 407,836, and filed on Sep. 15, 1989, now U.S. Pat. No. 5,325,478;

Method for Storing a Transaction in a Distributed Database System, having Ser. No. 408,164, and filed on Sep. 15, 1989, now abandoned;

A Method of Forming a Spreadsheet Display, having Ser. No. 407,972, and filed on Sep. 15, 1989, now abandoned;

Data Storage Audit Trail, having Ser. No. 409,230, and filed on Sep. 15, 1989, now abandoned;

Method for Handling Recursive Calculations, having Ser. No. 07/435,396 and filed on Nov. 13, 1989, now abandoned;

Attribute Import Between Nodes of a Directed Graph, having Ser. No. 07/443,616 and filed on Nov. 30, 1989, now abandoned;

Attribute Export Between Nodes of a Directed Graph, having Ser. No. 07/443,621 and filed on Nov. 30, 1989, now abandoned;

Association of Edit Functions With Attributes, having Ser. No. 07/443,623 and filed on Nov. 30, 1989, now abandoned;

Association of Edit Functions With Attribute Types, having Ser. No. 07/443,613 and filed on Nov. 30, 1989, now abandoned;

Association of Default Functions With Attributes, having Ser. No. 07/443,629 and filed on Nov. 30, 1989, now abandoned;

Association of Calculation Functions With Attributes, having Ser. No. 07/443,628 and filed on Nov. 30, 1989, now abandoned;

Association of List Functions With Attributes, having Ser. No. 07/443,622 and filed on Nov. 30, 1989, now abandoned;

Association of Cross-Field Edit Functions With Attributes, having Ser. No. 07/443,614 and filed on Nov. 30, 1989, now abandoned; and

Association of Combinations of Default, Edit, Calculation, and List Functions With Attributes, having Ser. No. 07/443,630 and filed on Nov. 30, 1989, now abandoned.

TECHNICAL FIELD

This invention relates generally to data processing systems and processes, and, in particular, to a data processing system in which various edit procedures may be enabled in a predetermined, yet modifiable order as data is entered into one or more fields of a form.

PROBLEMS ADDRESSED BY THE INVENTION

The present invention has utility in data processing systems in which various routine procedures are invoked in response to user input. The user input may take the form of one or more "attributes", i.e. the smallest piece of useful data in a data object (i.e. data field in the database). Refer to Glossary appended hereto as Appendix A.

Traditionally, libraries are built for routine procedures (such as defaults, edits, calculations, and lists, which hereinafter may be referred to DECL's). Each function is written to process one particular attribute. All applications must call the appropriate functions for the attributes they use. When a library function is changed, added, or deleted, all applications that use the function (i.e. the attribute the function is associated with) must be correctly modified, recompiled, and retested.

Using the present invention, routine functions are associated with attributes and are automatically invoked by a separate process. The application never deals with them. Also routine functions can be associated with more than one attribute or an attribute type, so that they can be reused.

As a result, responsibility for routine tasks is removed from the applications. There is a corresponding reduction in errors, maintenance time, implementation time, and design time. There is also a reduction in complexity, because DECL's are reuseable (can be used with more than one attribute), so not as many are needed. A library of nodes, or program parts, is built up as applications are constructed. In addition, the code is more readable and understandable.

By reducing the number of DECL's procedures in a system, maintainability is enhanced, and the implementation of applications is speeded up. Instead of writing new DECL's procedures, the applications designer can reuse existing ones. This increases programmer productivity, since the applications programmer is insulated from details he or she doesn't need to know.

Further, the present invention enhances the ability to customize and configure systems for different customer installations.