Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2443454 A1 2005/03/11

(21) 2 443 454

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(22) Date de dépôt/Filing Date: 2003/09/11

(41) Mise à la disp. pub./Open to Public Insp.: 2005/03/11

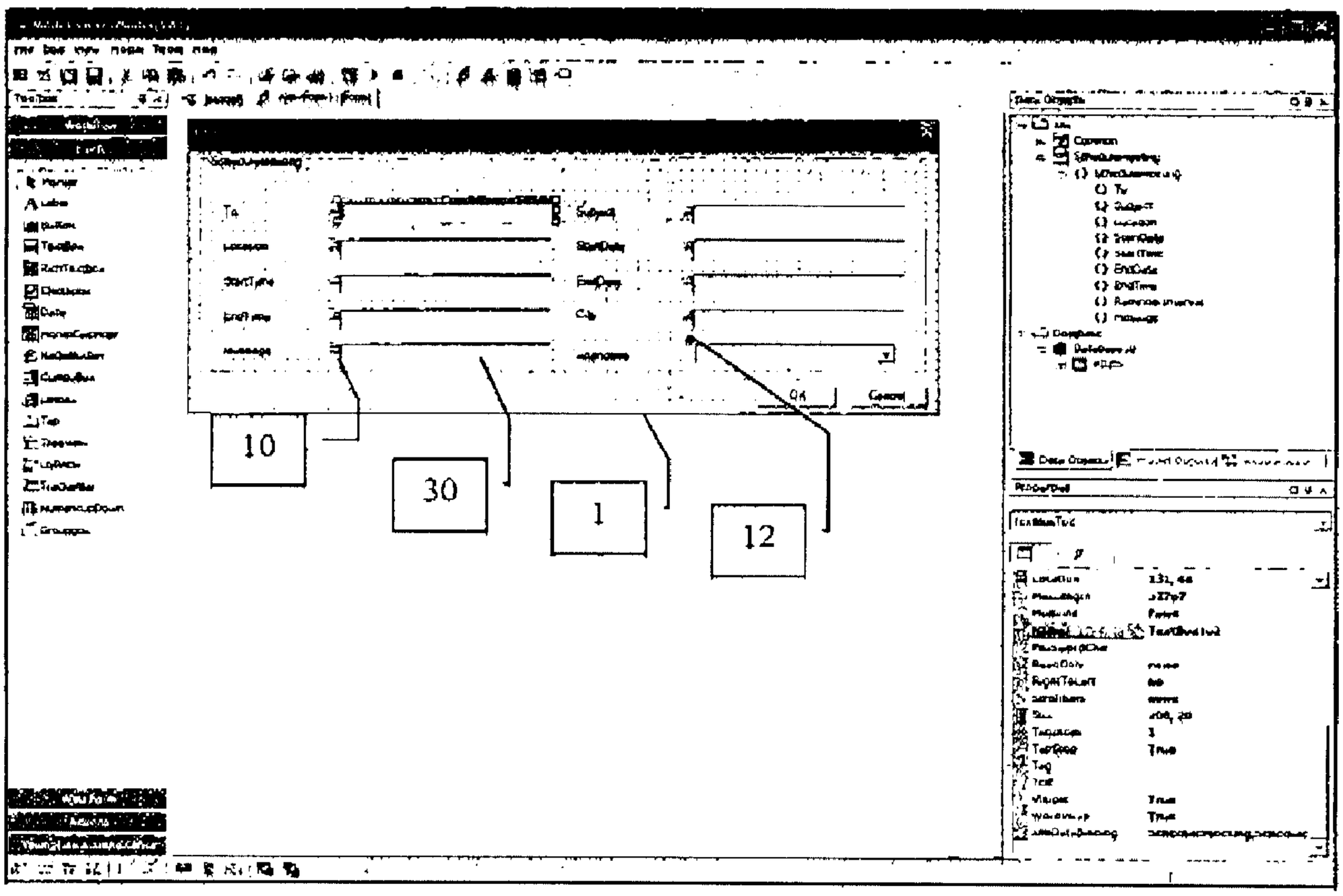
(51) Cl.Int.^{7}/Int.Cl.^{7} G06F 17/00, G06F 17/30, G06F 17/20

(71) Demandeur/Applicant:
TEAMPLATE INC., CA

(72) Inventeurs/Inventors:
TATTRIE, SCOTT I., CA;
DINN, TRENT SHELDON, CA

(74) Agent: BORDEN LADNER GERVAIS LLP

(54) Titre : METHODE D'ASSOCIATION DE DONNEES DANS UN SYSTEME D'ACHEMINEMENT DU TRAVAIL
(54) Title: DATA BINDING METHOD IN WORKFLOW SYSTEM



(57) Abrégé/Abstract:
A method for data binding a control to a data object by using drag and drop functionality without requiring additional programming. More particularly, the present system relates to either simple or complex data binding of a control in a form or a webform to an XML data object, database or other data object or source through drag and drop functionality in a graphical user interface without requiring a programmer to programmatically bind the control to the XML data object, database or other data object or source.

ABSTRACT

A method for data binding a control to a data object by using drag and drop functionality without requiring additional programming. More particularly, the present system relates to either simple or complex data binding of a control in a form or a webform to an XML data object, database or other data object or source through drag and drop functionality in a graphical user interface without requiring a programmer to programmatically bind the control to the XML data object, database or other data object or source.

DATA BINDING METHOD IN WORKFLOW SYSTEM

FIELD OF THE INVENTION

The present system relates generally to data binding a control to an XML data object or a database by using drag and drop functionality without requiring additional programming. More particularly, the present system relates to either simple or complex data binding of a control in a form or a webform to an XML data object, database or other data source through drag and drop functionality in a graphical user interface without requiring a programmer to programmatically bind the control to the data source.

BACKGROUND OF THE INVENTION

It is common when developing desktop or internet (web) applications to access data stored in an independent data source. One method of accessing data from a data source is data binding.

Data binding is generally the process of retrieving data from a data source and subsequently allowing such retrieved data to be used in a control. In particular, the retrieved data may be used to set a property of the control, which may, for example, include labelling the control (simple data binding) or it may be used to provide the contents for a control such as a listbox at run time (complex data binding).

Data binding also allows a programmer to insulate the data from the presentation layer of the application as the data is provided in an independent data source and the data's integrity and/or secrecy can therefore be maintained.

While data binding can be advantageous, due to limitations in previous application programming interfaces (APIs), data binding was awkward and it was difficult for programmers to control how data was bound to the control. More recent APIs have dramatically improved the effectiveness of data binding and have further included wizards to try to make data binding a more accessible system for programmers. However, these recent APIs still do not take advantage of drag and drop functionality to bind a control to data found in an independent data source which would improve the speed with which applications could be developed by reducing the need for additional programming.

It is therefore desirable to provide a data binding method using drag and drop functionality in a novel way to provide for more rapid application development.

SUMMARY OF THE INVENTION

It is an object of the present system to obviate or mitigate at least one disadvantage of previous data binding methods by introducing an aspect of drag and drop functionality.

Other aspects and features of the present system will become apparent to those ordinarily skilled in the art upon review of the following description of specific embodiments of the system in conjunction with the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the present system will now be described, by way of example only, with reference to the attached Figures, wherein:

Fig. 1 is a screen shot of the design view of a form showing a hierarchical list of XML object elements and nodes in accordance with the present invention;

Fig. 2 is a screen shot of the design view of a form showing a hierarchical list of database object elements and nodes in accordance with the present invention;

Fig. 3 is a control on a form in design view showing an iconic indicator of data binding to an XML object;

Fig. 4 is a control on a form in design view showing an iconic indicator of data binding to an database object; and

Fig. 5 is a form in design view populated with controls which are data bound to XML objects.

DETAILED DESCRIPTION

Generally, the present system provides a data binding method using drag and drop functionality. More specifically, the present system relates to either simple or complex data binding of a control in a form or webform to an XML data object, database or other data source through drag and drop functionality in a graphical user interface without requiring a programmer to programmatically bind the control to the data source. By binding controls to an element or node in an XML data object or by binding the controls to a database or other data source, the XML variables or database variables, respectively, are loaded to the controls; further, on closing the forms containing the controls, the control's properties are loaded to the XML variables or database variables, respectively.

XML data binding is performed by first displaying the design view or design layer of a form or webform and exposing the XML data object to reveal the XML data object elements and nodes, as shown in Figure 1. The XML data object elements and nodes may be displayed hierarchically in a tree view or as a list. Next, the programmer drags and drops one of the elements or nodes onto the required control type which may be a label, button, textbox, checkbox, data, radio button, combo box or listbox, depending on the development requirements and depending on whether the programmer requires simple or complex data binding.

Once the XML element or node is dropped on the control type, an icon may appear which indicates that the control is bound to an XML element or node.

Next, the programmer drags and drops the control 30 onto the form 1 or webform, as necessary, at which point the data binding process is complete. An icon 10 or specific colour or other property may indicate to the programmer that the control is bound to an XML element or node.

After completing the data binding process, the programmer may determine which XML element or node a particular control 30 is bound to by checking a property of the control 30, typically the XMLDataBinding property. This property may provide information such as the name of the XML data object and the path and name of the element or node. From this property view, the programmer may also update, change or modify the XML data binding to another element or node, as required. The programmer may also delete the XML data binding from the property view or, alternatively, a shortcut may be provided upon right-clicking or other invocation step on a control 30 which allows the data binding to be removed.

The present invention may allow a programmer to quickly create an entire form or webform by repeating the steps above. However, the present invention also may allow a programmer to select an XML data object and invoke a shortcut (such as right-clicking on the XML data object) to create the entire form or webform with a control automatically created for each element or node, or, alternatively, invoke a shortcut for specific nodes to populate the form or webform.

The above advantages can also be attained by data binding a control 30 to a database data object or even a data object's output rather than an XML data object, thereby greatly

increasing the potential uses of the present invention. A control 30 bound to a database data object is shown in Figure 2 and Figure 4 (and can be discerned as a control bound to a database data object by icon 12).

In addition to alleviating the need for the programmer to programmatically data bind the control 30 to the XML data object, the present invention further provides the advantage that the necessary script or programming is also completed for the form or page load event for the form or webform, respectively, and, further, the present invention automatically completes the necessary code for the click event of the OK button (or submit button 2) on the form (or webform) based on the data binding completed previously.

The above-described embodiments of the present system are intended to be examples only. Alterations, modifications and variations may be effected to the particular embodiments by those of skill in the art without departing from the scope of the system, which is defined solely by the claims appended hereto.

CLAIMS:

1. A method for data binding data a control to a data source comprising the steps of:
   a. exposing the nodes of the data source;
   b. dragging and then subsequently dropping a node from the data source onto an icon representing the control; and
   c. dragging the cursor from the control to a form and dropping the control on the form.

2. A method for data binding data a control to an XML data object comprising the steps of:
   a. exposing the nodes of the XML data object within the design view of a form;
   b. dragging and then subsequently dropping a node from the XML data object onto an icon representing the control; and
   c. dragging the cursor from the control to the form and dropping the control on the form.

FIGURE 1

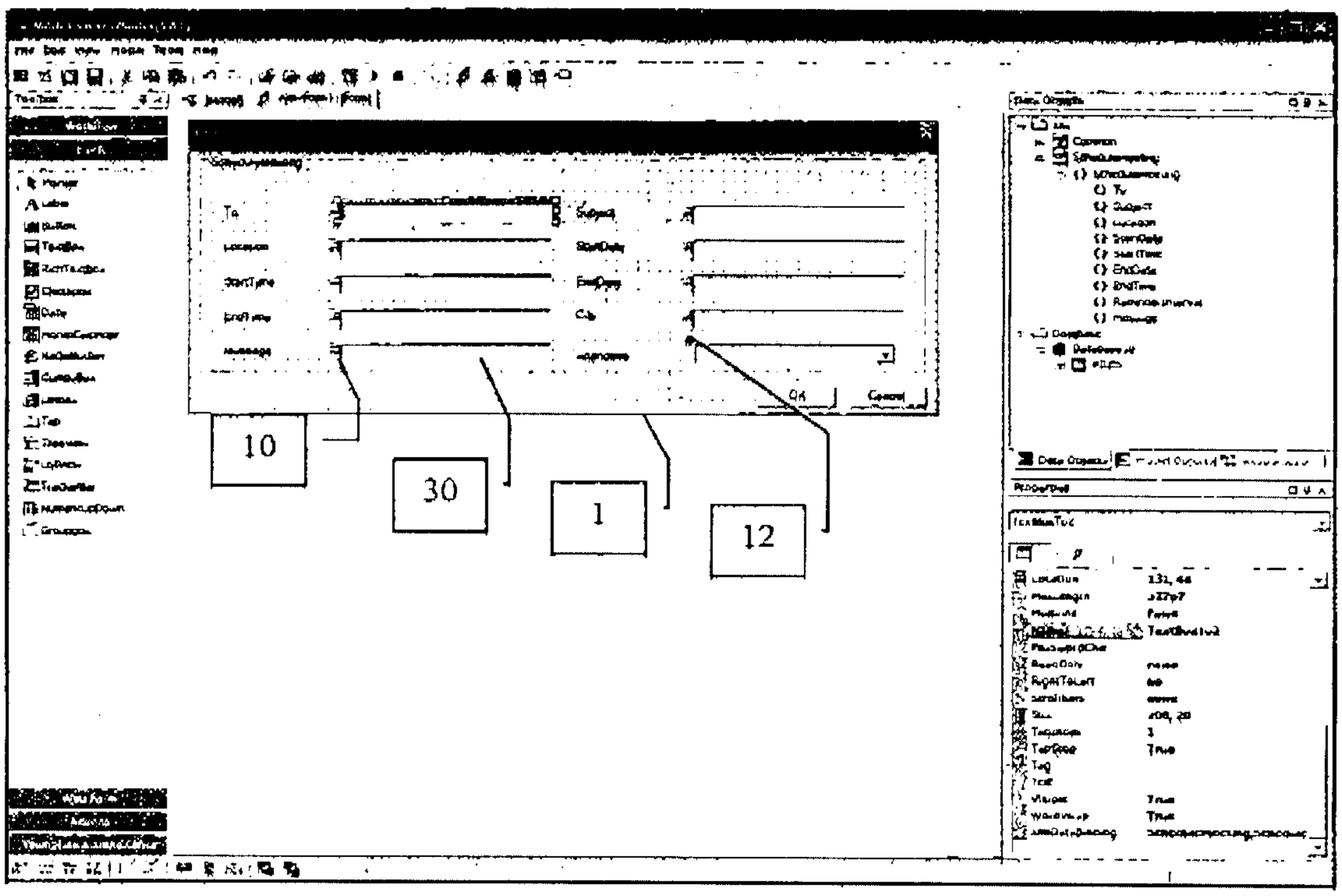


FIGURE 2

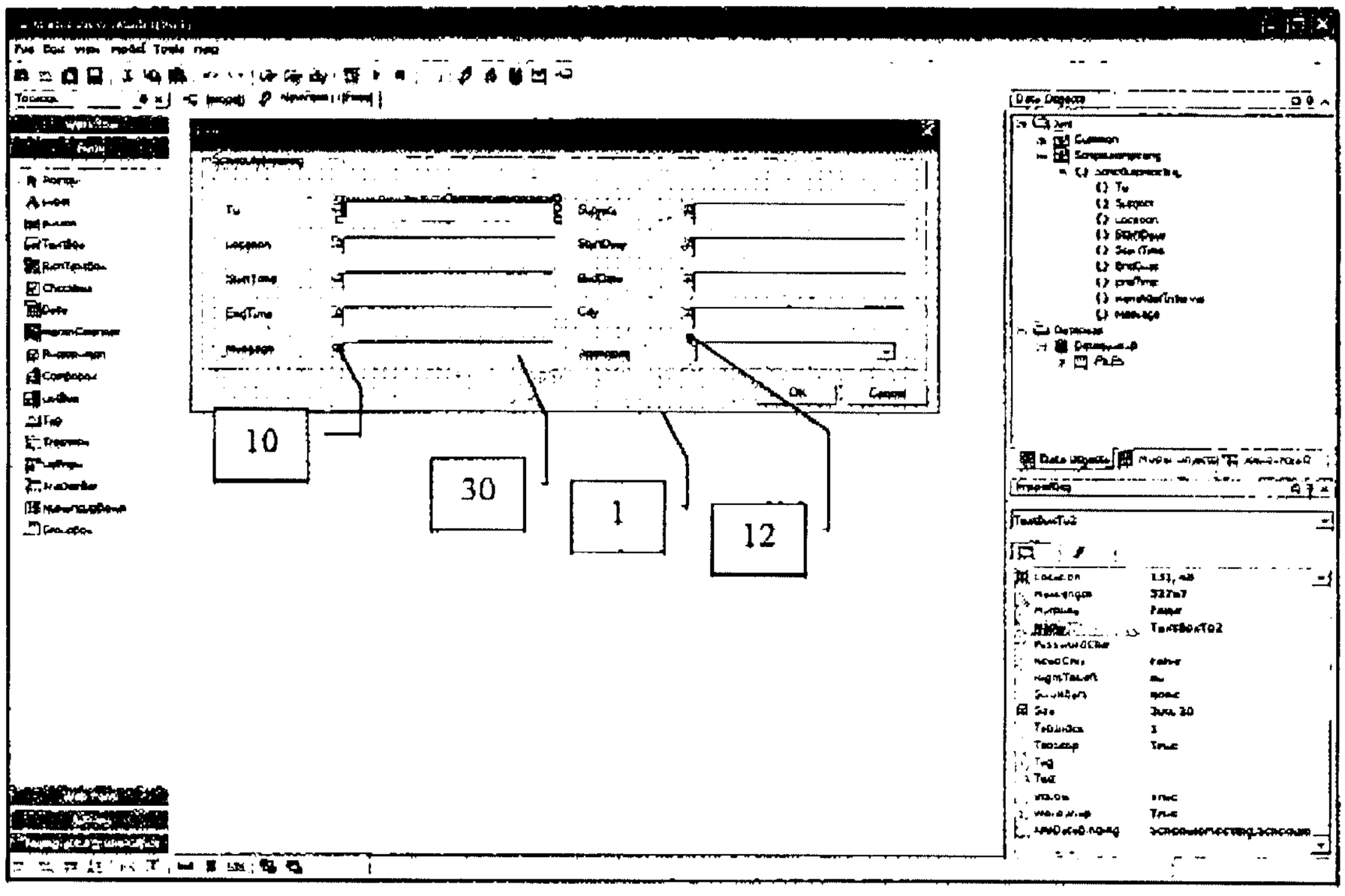


FIGURE 3

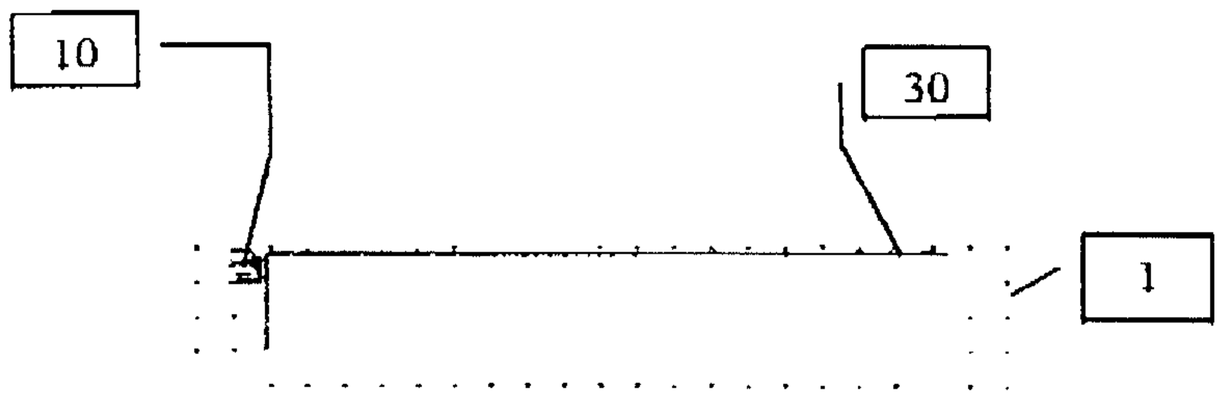


FIGURE 4

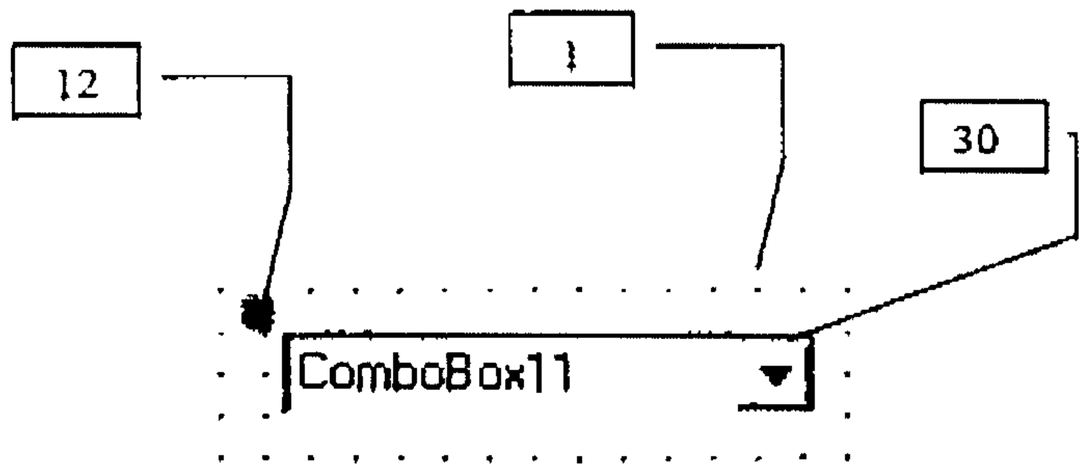
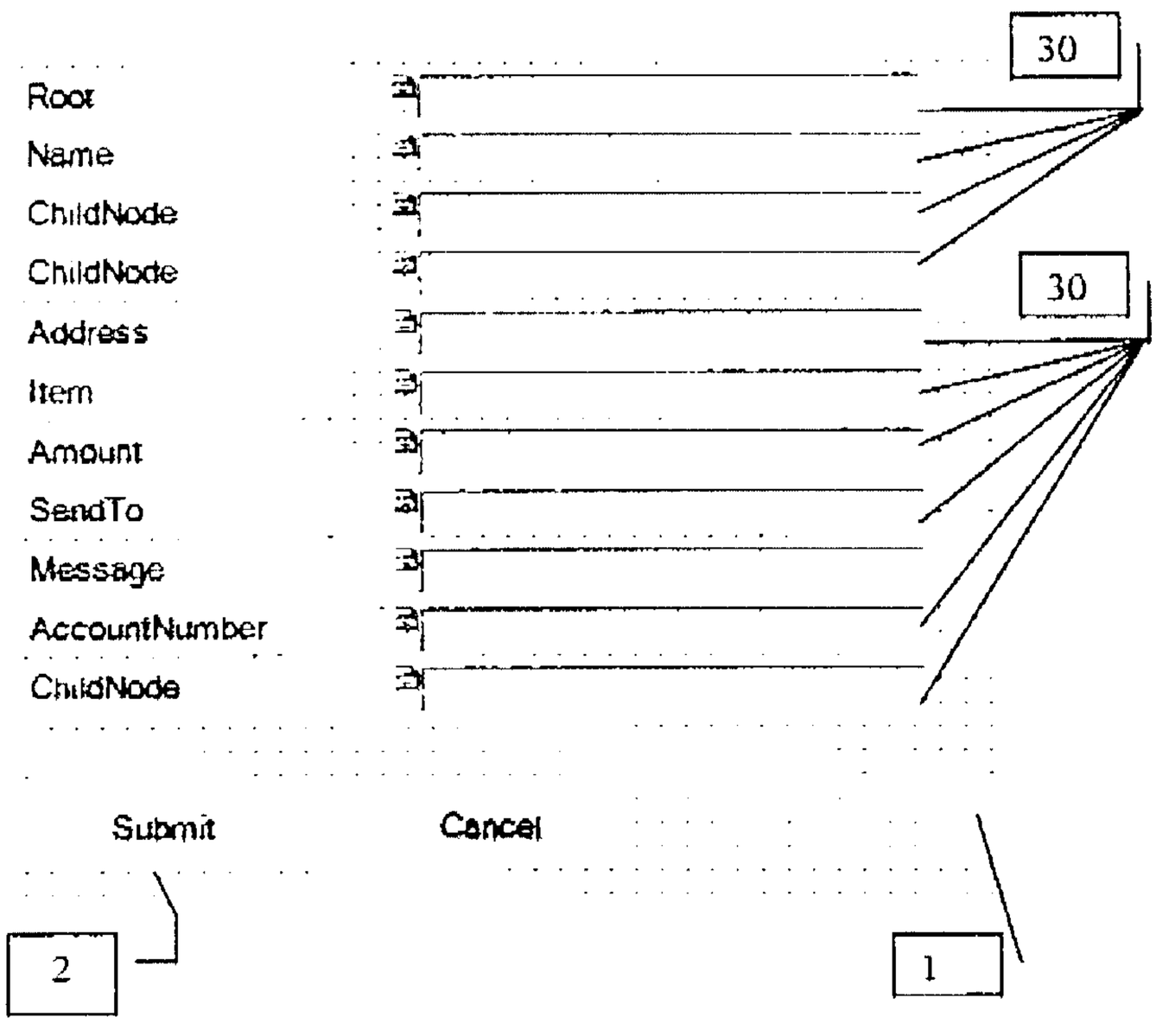
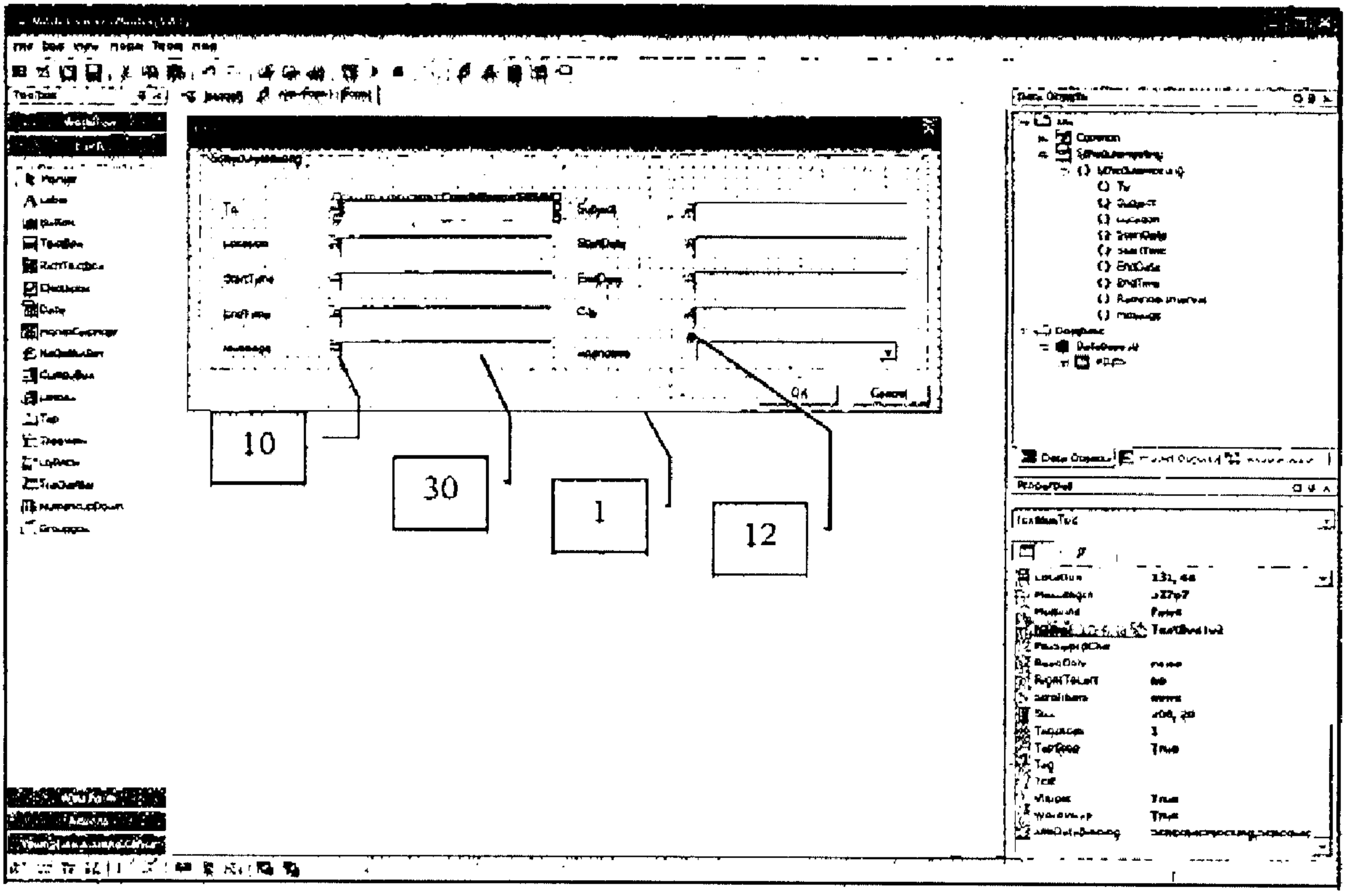


FIGURE 5


Root
Name
ChildNode
ChildNode
Address
Item
Amount
SendTo
Message
AccountNumber
ChildNode
30
30
Submit
Cancel
2
1


10
30
1
12