

creating at least one descendant form, by inheriting said set of components from the ancestor form;
 modifying said at least one descendant form by overriding a value stored for at least one property;
 storing said at least one descendant form by storing values for only those properties which have been overridden;
 creating a new component at the ancestor form; and
 in response to said creating step, creating at each descendant a corresponding new component.

19. The method of claim 18, wherein each corresponding component created at a descendant initially includes properties having values identical to values of corresponding properties of the ancestor component.

20. In a development system having a processor and a memory, a method for assisting a user with creating an application program, the method comprising:

creating a first form having a first component, said first component being associated with an event handler for processing events which occur at said first component; and

creating a second form which inherits from said first form, said second form having a second component which inherits from said first component, said second component automatically being associated with said event handler for processing events which occur at said second component;

wherein said event handler includes programming code which executes upon invocation of the event handler, wherein a component is associated with a particular event handler by storing at that component a reference to that event handler, and wherein said reference includes a function pointer which points to a location in said memory where said program code of the event handler resides.

21. The method of claim 20, wherein said event handler includes programming code which executes upon invocation of the event handler.

22. The method of claim 20, wherein a component is associated with a particular event handler by storing at that component a reference to that event handler.

23. The method of claim 20, wherein said first component comprises a user interface component which can receive user input.

24. The method of claim 20, wherein said first component comprises a user interface component and wherein said second component comprises a user interface component having at least some properties which are identical to that of the first component.

25. The method of claim 20, further comprising:

overriding the event handler which is automatically associated with said second component, in response to user input assigning a new event handler to said second component.

26. In a development system having a processor and a memory, a method for assisting a user with creating an application program, the method comprising:

creating a first form having a first component, said first component being associated with an event handler for processing events which occur at said first component; and

creating a second form which inherits from said first form, said second form having a second component which inherits from said first component, said second com-

ponent automatically being associated with said event handler for processing events which occur at said second component;

wherein said event handler includes programming code which executes upon invocation of the event handler, wherein a component is associated with a particular event handler by storing at that component a reference to that event handler, and wherein said reference includes a "this" pointer which references a particular form in memory, said particular form having a particular component where an event occurs giving rise to invocation of the event handler.

27. The method of claim 26, wherein each form is represented in the system as an object instance created from a form class, and wherein said "this" pointer comprises a pointer to the object instance for the particular form.

28. In a development system having a processor and a memory, a method for assisting a user with creating an application program, the method comprising:

creating a first form having a first component, said first component being associated with an event handler for processing events which occur at said first component; and

creating a second form which inherits from said first form, said second form having a second component which inherits from said first component, said second component automatically being associated with said event handler for processing events which occur at said second component;

wherein said event handler is automatically associated with said second component of said second form by assignment of a method pointer into a handler of the second component, said assigned method pointer comprising a "this" pointer referencing said second form together with a function pointer referencing a location in said memory where programming code for the event handler resides.

29. A development system comprising:
 a computer having a processor and a memory;
 an interface for creating form objects having components; and,

means for creating a descendant form object from an existing ancestor form object, said means including means for automatically propagating to each descendant form object so created modifications which occur at said ancestor form object;

wherein said means for automatically propagating includes means for notifying a descendant form object that its corresponding ancestor form object has been modified.

30. The system of claim 29, wherein each descendant form object created comprises components having at least some properties which are identical to properties of corresponding components of the ancestor form object.

31. The system of claim 30, wherein each component has a name which uniquely identifies that component.

32. The method of claim 29, wherein said means for automatically propagating includes:

means for filtering properties of the corresponding ancestor form object so that only properties which have not been overridden at the modified descendant form object are propagated.