

receiving an expression into a property input section of the property entry sheet, the expression being referenceable at run-time as a data value;

parsing the expression;

generating from the expression a function; and

storing the function as a run-time value.

12. The method of claim 11, wherein the object has a byte code execution image, further comprising invalidating the byte code execution image.

13. The method of claim 11, further comprising determining whether a run-time display of the object is automatically updated, and if so, generating and executing the byte code.

14. The method of claim 11, further comprising changing an attribute of the object.

15. The method of claim 14, further comprising:

determining which object property maps to the changed attribute;

creating a constant function representing the changed attribute value;

storing the constant function as a run-time property value and a design-time property value.

16. The method of claim 14, wherein the object has a byte code execution image, further comprising invalidating the byte code execution image.

17. The method of claim 14, further comprising determining whether a run-time display of the object is automatically updated, and if so, generating and executing the byte code.

18. The method of claim 11, further comprising cloning and storing the function as a design time value if the function is a constant.

19. The method of claim 11, further comprising displaying an error message if the expression is invalid.

20. The method of claim 11, further comprising invalidating an existing byte code execution image and replacing the existing byte code execution image with a new byte code execution image associated with the function.

21. A computer implemented method for assigning a property to an object, the object having a value input adapted to receive a functional expression for the property, the functional expression being referenceable at run-time as a data value, the method comprising:

receiving an expression for the property into the value input;

parsing the expression and storing the expression as a function; and

evaluating the expression at run-time in context to generate a referenceable data value for the property.

22. The method of claim 21, wherein the parsing the expression identifies one or more of the following:

a function;

an operator;

a database column name;

a variable; and

a constant.

23. The method of claim 21, further comprising:

receiving an identification of the property in an attribute name section; and

receiving a static data value for the property identified by the attribute name section in a property input section.

24. The method of claim 21, wherein the object has a plurality of properties and wherein the attribute name section and the property input section of each property form a name-value pair for each property.

25. The method of claim 21, further comprising parsing the functional expression to generate a function which is stored as a run-time value.

26. The method of claim 25, further comprising associating a byte code with the function.

27. The method of claim 24, further comprising determining whether a run-time display of the object is automatically updated, and if so, generating and executing the byte code.

28. The method of claim 21, further comprising cloning and storing the function as a design time value if the function is a constant.

29. The method of claim 21, further comprising displaying an error message if the expression is invalid.

30. The method of claim 21, further comprising invalidating an existing byte code execution image and replacing the existing byte code execution image with a new byte code execution image associated with the function.

31. A computer-implemented object, comprising:

an object state; and

one or more interfaces providing access to the object state through a plurality of attributes, each of the attributes defined as a functional expression and referenceable at run-time as a data value.

32. The object of claim 31, wherein the functional expression includes one or more of the following:

a function;

an operator;

a database column name;

a variable; and

a constant.

33. The object of claim 31, wherein the attribute is a static data value.

34. The object of claim 31, further comprising a name-value pair for each property.

35. The object of claim 31, wherein the functional expression is parsed to generate a function which is stored as a run-time value.

36. The object of claim 35, further comprising byte code associated with the function.

37. The object of claim 31, wherein the function is cloned and stored as a design time value if the function is a constant.

38. The object of claim 31, wherein an error message is displayed if the expression is invalid.

39. The object of claim 31, wherein an existing byte code execution image is invalidated.

40. The object of claim 39, wherein new byte code is generated to replace the existing byte code execution image.

41. A computer system, comprising:

a processor;

a memory array coupled to said processor;