



US006678887B1

(12) **United States Patent**
Hallman et al.

(10) **Patent No.:** **US 6,678,887 B1**
(45) **Date of Patent:** **Jan. 13, 2004**

(54) **CUSTOMIZING BUSINESS LOGIC AND DATA SOURCES BY MODIFYING METHODS DEFINED WITHIN AN API**

(75) Inventors: **Clinton J. Hallman**, San Jose, CA (US); **David Forney**, Cupertino, CA (US)

(73) Assignee: **Networks Associates Technology, Inc.**, Santa Clara, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 507 days.

(21) Appl. No.: **09/613,653**

(22) Filed: **Jul. 11, 2000**

(51) **Int. Cl.**⁷ **G06F 9/44**

(52) **U.S. Cl.** **717/162; 717/171; 717/176**

(58) **Field of Search** **717/162, 171, 717/176; 707/171**

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,021,410 A * 2/2000 Choy 707/103 R
6,101,502 A * 8/2000 Heubner et al. 707/103 R
6,182,142 B1 * 1/2001 Win et al. 709/229
6,321,374 B1 * 11/2001 Choy 717/106
6,412,071 B1 * 6/2002 Hollander et al. 713/200

OTHER PUBLICATIONS

IBM Dictionary of Computing, Geroge Mc Daniel, p. 255, published 1994.*

IBM Technical Disclosure Bulletin, vol 31, no 1, Jun. 1988, pp. 294-298.*

* cited by examiner

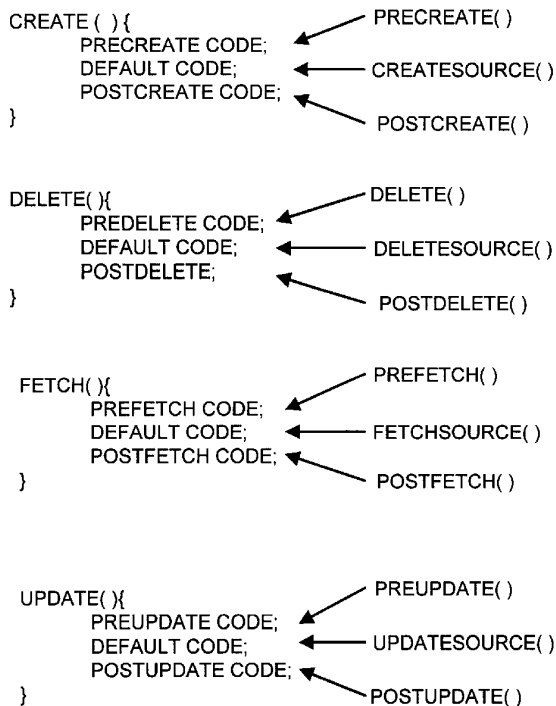
Primary Examiner—Todd Ingberg

(74) *Attorney, Agent, or Firm*—Silicon Valley IP Group, PC; Kevin J. Zilka; Christopher J. Hamaty

(57) **ABSTRACT**

A system that facilitates customizing a software package by modifying an implementation of a target method defined within an application programming interface (API) for the software package is presented. The system operates by receiving additional code for integration into a target method defined within the API and a command to integrate this code, wherein the API defines a plurality of methods that operate on objects. This command is received through a pre-defined method within the API. In response, the system links the additional code into the target method so that the additional code is executed upon invocation of the target method. In one embodiment the API defines: a method that creates an object; a method that deletes the object; a method that fetches the object; and a method that updates the object. In one embodiment the additional code causes the target method to operate on data from an alternative source.

24 Claims, 3 Drawing Sheets



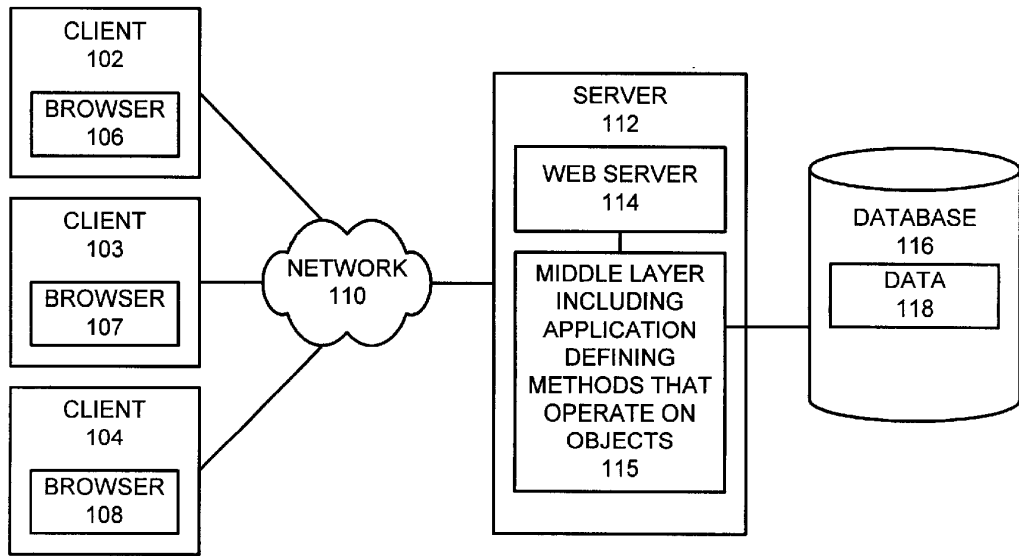


FIG. 1

DISTRIBUTED COMPUTING SYSTEM 100

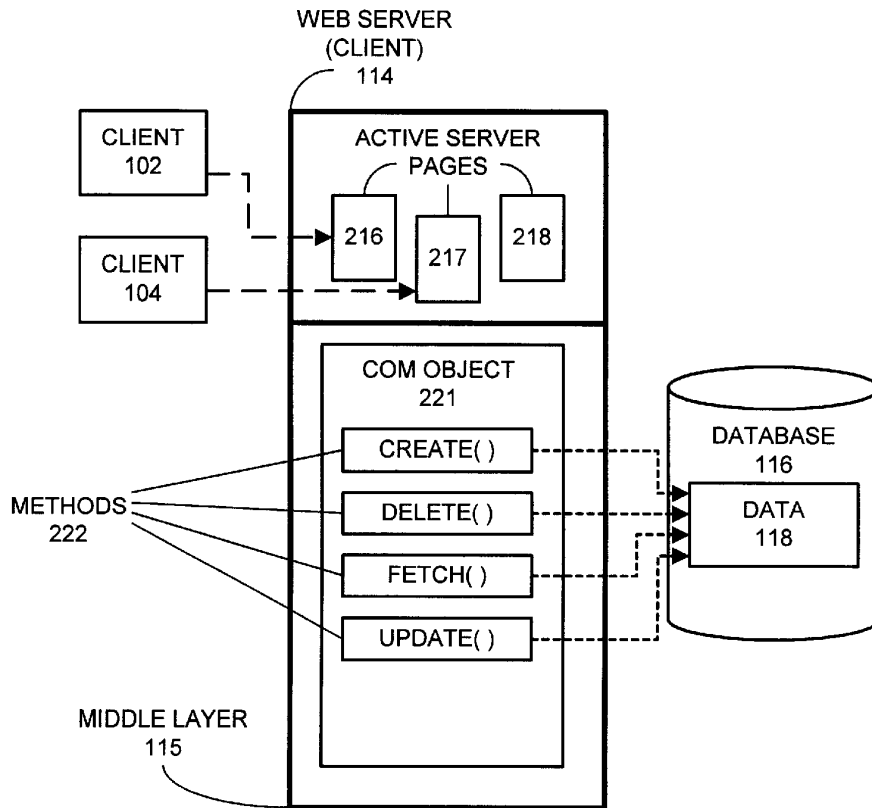


FIG. 2

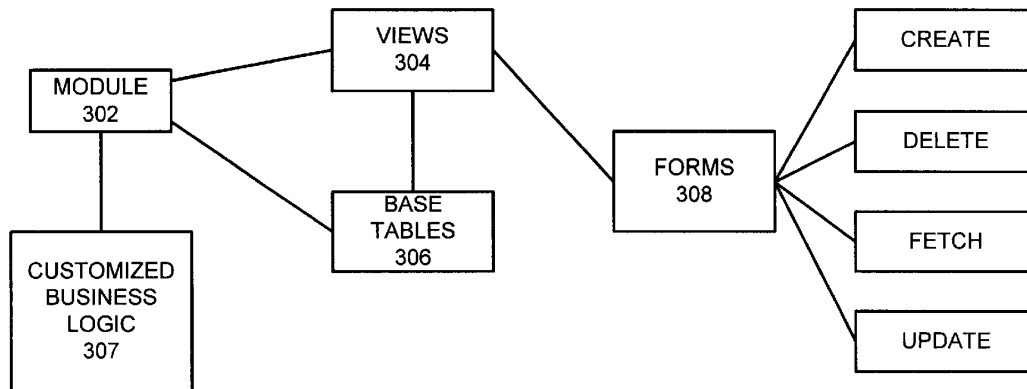


FIG. 3

```

CREATE ( ) {
    PRECREATE CODE; ← PRECREATE ( )
    DEFAULT CODE; ← CREATESOURCE ( )
    POSTCREATE CODE; ← POSTCREATE ( )
}

DELETE ( ) {
    PREDELETE CODE; ← DELETE ( )
    DEFAULT CODE; ← DELETESOURCE ( )
    POSTDELETE; ← POSTDELETE ( )
}

FETCH ( ) {
    PREFETCH CODE; ← PREFETCH ( )
    DEFAULT CODE; ← FETCHSOURCE ( )
    POSTFETCH CODE; ← POSTFETCH ( )
}

UPDATE ( ) {
    PREUPDATE CODE; ← PREUPDATE ( )
    DEFAULT CODE; ← UPDATESOURCE ( )
    POSTUPDATE CODE; ← POSTUPDATE ( )
}
    
```

FIG. 4

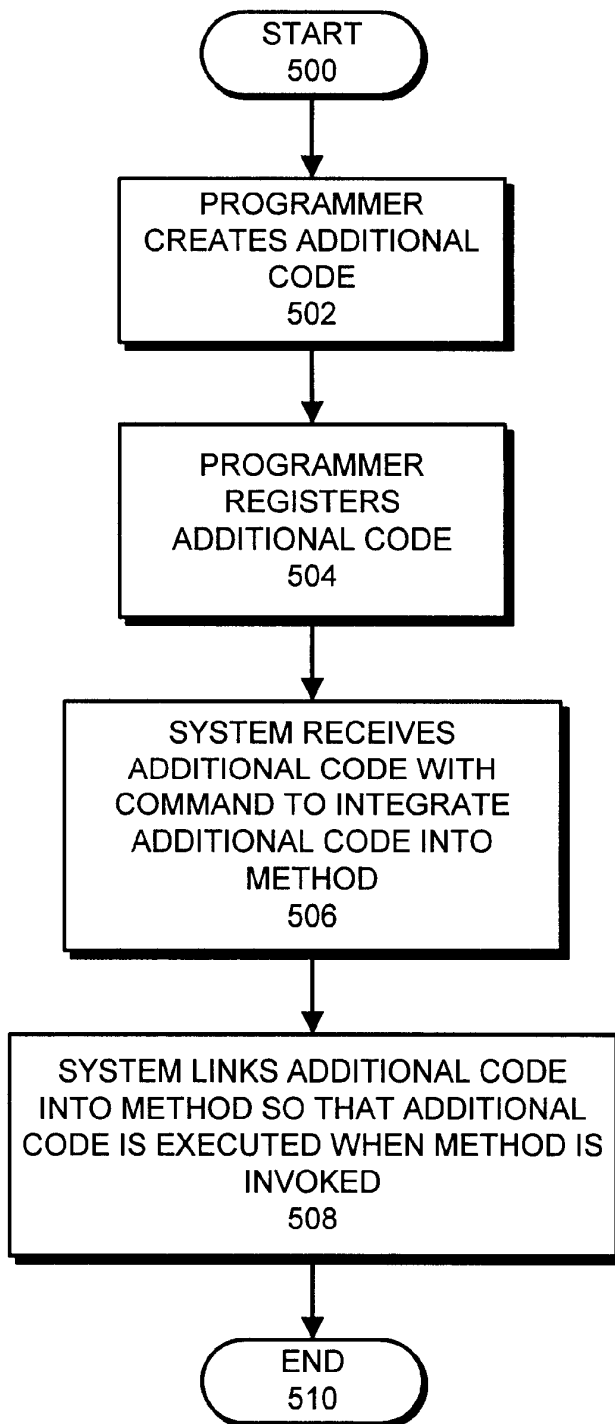


FIG 5.

**CUSTOMIZING BUSINESS LOGIC AND
DATA SOURCES BY MODIFYING METHODS
DEFINED WITHIN AN API**

BACKGROUND

1. Field of the Invention

The present invention relates to application programming interfaces (APIS) for computer systems. More specifically, the present invention relates to a method and an apparatus for modifying an application by customizing methods defined within an API for the application to include additional business logic and/or to specify additional data sources.

2. Related Art

As businesses scramble to exploit the potential efficiencies that arise through the use of information technology, numerous business processes are becoming computerized. This trend has led to an explosion in the development of software packages to facilitate specific business processes. Unfortunately, organizations are typically unable to make use of off-the-shelf applications to facilitate a specific business process because the requirements of a specific business process are typically dictated by requirements of a specific type of business and a specific organizational structure. Hence, a business is typically forced to develop its own software package for a specific business process. This typically entails employing expensive teams of programmers and/or consultants to develop, debug and maintain the software package.

The process of developing a software package can consume a great deal of time and can distract management from focussing on the main line of a business. Furthermore, a software package must be continually maintained and updated, which can require additional commitments of time and financial resources.

For example, a company may want to design a software package to facilitate a "help desk" for computer system support. A completely off-the-shelf help desk system is typically not practical to use because different organizations have different requirements. For example, one company may want to tie a help desk system into a specific inventory database to keep track of parts that are used in fixing a computer system. Another company may want to use the help desk system to facilitate automatic purchasing for parts. Yet another company may want to use the help desk system to track equipment faults.

However, note that building a completely new software system from the ground up for each company is wasteful because many elements of a help desk system can be shared between different organizations.

Hence, what is needed is a method and an apparatus that facilitates customizing an off-the-shelf software system to fulfill specific requirements of a specific business process for a specific organization.

SUMMARY

One embodiment of the present invention provides a system that facilitates customizing a software package by modifying an implementation of a target method defined within an application programming interface for the software package. The system operates by receiving additional code to be integrated into a target method defined within the application programming interface, wherein the application programming interface defines a plurality of methods that

operate on objects. The system also receives a command to integrate the additional code into the target method within the application programming interface. This command is received through a pre-defined code integration method within the application programming interface. In response to this command, the system links the additional code into the target method within the application programming interface so that the additional code is executed when the target method is invoked.

In one embodiment of the present invention, the application programming interface defines: a method that creates an object; a method that deletes the object; a method that fetches the object; and a method that updates the object.

In one embodiment of the present invention, the object includes: an object associated with a help desk; an object associated with a client of the help desk; an object associated with a support staff member associated with the help desk; an object associated with an inventory item for the help desk; and an object associated with a subject of a help desk ticket.

In one embodiment of the present invention, the additional code causes the target method to operate on data from an alternative data source.

In one embodiment of the present invention, the pre-defined code integration method integrates the additional code so that it is executed prior to executing default code for the target method.

In one embodiment of the present invention, the pre-defined code integration method integrates the additional code so that it is executed after executing default code for the target method.

In one embodiment of the present invention, the application programming interface is implemented in a middle tier of a three-tier distributed computing architecture, which includes: a client tier for interacting with users; the middle tier for performing business logic functions; and a database tier for storing data.

In one embodiment of the present invention, the application programming interface is defined within a common object model (COM) object.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 illustrates a distributed computing system in accordance with an embodiment of the present invention.

FIG. 2 illustrates the structure of a web server and a middle layer in accordance with an embodiment of the present invention.

FIG. 3 illustrates structures related to a module in accordance with an embodiment of the present invention.

FIG. 4 illustrates how several methods are implemented so they can be modified in accordance with an embodiment of the present invention.

FIG. 5 is a flow chart illustrating the processing of incorporating additional code into a method in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.