(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103809983 A
(43)申请公布日 2014.05.21

(21)申请号 201410068650.X

(22)申请日 2014.02.27

(71)申请人 山东超越数控电子有限公司
地址 250100 山东省济南市高新区孙村镇科航路2877号

(72)发明人 鄢建龙

(51)Int.Cl.
G06F 9/44(2006.01)
G06F 9/45(2006.01)

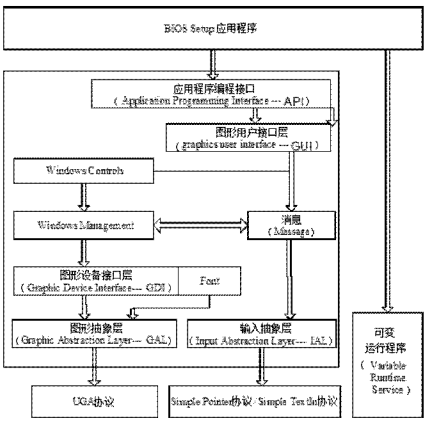
权利要求书2页 说明书6页 附图1页

(54)发明名称

一种修改BIOS SETUP界面的方法

(57)摘要

本发明公开了一种修改BIOSSETUP界面的方法，属于计算机UEFI技术领域。计算机系统在UEFIShell环境中，所述方法使用的系统架构包括提供图形接口的图形抽象层、提供鼠标和键盘驱动的输入抽象层、图形设备接口层、图形用户接口层、应用程序编程接口，图形抽象层和输入抽象层位于系统最底层，图形设备接口层位于系统的中间层，图形用户接口层位于图形设备接口层上方，系统的最顶层为应用程序编程接口。本发明大大降低了系统的耦合度，提高了系统的可扩展性；使系统取得了较好的交互体验和较美观的界面效果。



CN 103809983 A

1. 一种修改BIOS SETUP界面的方法，其特征在于计算机系统在UEFI Shell环境中，所述方法使用的系统架构包括提供图形接口的图形抽象层、提供鼠标和键盘驱动的输入抽象层、图形设备接口层、图形用户接口层、应用程序编程接口，图形抽象层和输入抽象层位于系统最底层，图形设备接口层位于系统的中间层，图形用户接口层位于图形设备接口层上方，系统的最顶层为应用程序编程接口；

与图形抽象层和输入抽象层相对应的接口部分是UEFI中的UGA协议、Simple Pointer协议和Simple TextIn协议，其中系统的图形抽象层对UGA协议进行封装，并为上层的图形设备接口层提供统一的接口，图形设备接口层则对图形抽象层进行封装，为上层的图形用户接口层提供功能使用；输入抽象层则对Simple Pointer协议与SimpleTextIn协议进行封装，并通过对鼠标键盘输入的监视生成消息，通过消息可以进行窗口之间的通信，为图形用户接口层的(窗口创建与绘制、多窗口之间的剪切等)的功能提供服务；图形用户接口层通过应用程序编程接口访问BIOS Setup的应用程序；

所述方法步骤为：

(1)、在UEFI的BIOS源代码中添加图形抽象层和输入抽象层的驱动并实现相关的UGA协议、Simple Pointer协议和Simple TextIn协议；

(2)、编译UEFI的BIOS源代码产生可执行的ROM文件；

(3)、更新步骤(2)中编译的BIOS源代码文件到计算机系统Firmware中；

(4)、编写UEFI应用程序实现BIOS Setup界面的显示和控制；

(5)、启动计算机系统配置文件到UEFI Shell模式下，运行应用程序。

2. 根据权利要求1所述的一种修改BIOS SETUP界面的方法，其特征在于步骤(1)的实现，将图形抽象层和输入抽象层的程序按照驱动协议模型编写；包括如下步骤：①、编写UGA协议、Simple Pointer协议和Simple TextIn协议的接口；②、用驱动格式实现上述接口。

3. 根据权利要求2所述的一种修改BIOS SETUP界面的方法，其特征在于步骤(1)的①编写协议部分，需要在BIOS的Protocol文件夹下建立图形抽象层或输入抽象层的文件夹，并在该文件夹中编写Protocol的接口函数；

步骤(1)的②驱动格式是在BIOS的Driver文件夹下建立图形抽象层或输入抽象层的文件夹，在这个文件夹下编写Protocol文件中GAL函数接口的具体实现；需要实现Supported函数、Start函数与Stop函数；

Supported函数实现步骤：用UEFI规范的OpenProtocol函数打开UEFI的UGA协议并加以测试，以保证之后的GAL功能函数可以调用UGA的服务，之后用UEFI规范的CloseProtocol函数关闭UGA协议；

Start函数实现步骤：在得到Supported函数的保证下，用OpenProtocol函数打开UEFI的UGA协议以调用其提供的服务；随后构造相应的数据结构并用之前Protocol中的接口函数指针指向驱动程序中对应的函数；最后用UEFI规范的InstallProtocolInterface函数将GAL协议安装到Controller Handle上；

Stop函数实现步骤：将Start中安装的协议用CloseProtocol关闭，并用EFI规范的UninstallProtocolInterface函数卸载，最后释放所构造的数据结构；

具备了上面3个函数的驱动程序就实现了UEFI驱动的绑定协议，为GAL功能函数

的开发提供了一个符合 UEFI 驱动模式的框架；

接下来用图形学的算法实现相应的函数接口；

最后，通过在 UEFI 应用程序中调用 GAL 和 IAL 的驱动来实现对 BIOS Setup 界面的显示和修改；这包括图形设备接口的实现、字符的显示、窗口的管理和控制操作；窗口的管理和控制通过消息传递的方式来实现，通过调用 IAL 的驱动产生消息传递给窗口控制程序；

对于 BIOS Setup 菜单的显示需要调用 UEFI 运行时服务的接口函数 GetVariable 和 GetNextVariableName 以及 QueryVariableInfo；在对 Setup 菜单的数值进行修改后需要调用 SetVariable 接口函数。

一种修改 BIOS SETUP 界面的方法

[0001]

技术领域

[0002] 本发明涉及一种计算机 UEFI 技术领域，具体地说是一种修改 BIOS SETUP 界面的方法。

背景技术

[0003] UEFI 为英文 Unified Extensible Firmware Interface 的缩写，翻译为统一的可扩展固件接口，是一种详细描述类型接口的标准。UEFI 是操作系统与平台固件之间的接口模型。UEFI 用于操作系统自动从预启动的操作环境，加载到一种操作系统上，为启动一个操作系统与执行启动前程序提供了一个标准环境。

[0004] 在计算机科学中，Shell 俗称壳(用来区别于核)，是指“提供使用者使用界面”的软件(命令解析器)。它接收用户命令，然后调用相应的应用程序。UEFI Shell 提供了类似 DOS 的命令行文字界面。

[0005] BIOS 是英文“Basic Input Output System”的缩略语，直译过来后中文名称就是“基本输入输出系统”。其实，它是一组固化到计算机内主板上一个 ROM 芯片上的程序，它保存着计算机最重要的基本输入输出的程序、系统设置信息、开机后自检程序和系统自启动程序。其主要功能是为计算机提供最底层的、最直接的硬件设置和控制。BIOS Setup 菜单提供了对系统的配置信息进行设置的人机图形界面。

[0006] 如何在 UEFI Shell 下修改 BIOS Setup 界面是本领域技术人员迫切需要解决的问题。

[0007] 发明内容

本发明的技术任务是针对以上不足之处，提供一种大大降低了系统的耦合度，提高了系统的可扩展性；使系统取得了较好的交互体验和较美观的界面效果的一种修改 BIOS SETUP 界面的方法。

[0008] 在 UEFI 规范中包括了 UGA(universal graphics adapter 通用的图形适配器）协议，该协议提供了一个 Blt 的函数接口在内存与设备之间进行像素数据的传输；另外用于鼠标控制的 Simple Pointer 协议与用于键盘输入的 Simple TextIn 协议提供了鼠标与键盘的交互功能。但是按照 UEFI 的规范，使用底层的设备需要按照协议－驱动模型结构编写代码，这样如果将来上述这 3 个协议产生了变化，也就需要同时对 GDI (英文全称 graphics device interface，翻译为图形设备接口层）中的绘图支持部分以及鼠标键盘部分进行相应的修改，进一步会牵涉到 GDI 中其他功能对绘图支持部分与鼠标键盘部分的调用，极大地增加了不稳定性。因此为了提高系统的可移植性，这 3 个协议的调用部分需要重新剥离出去生成新的层次，以满足 UEFI 协议－驱动模型的要求。对 CMOS 和变量区域的访问可以通过调用运行时服务的函数接口 GetVariable 和 SetVariable 以及 QueryVariableInfo 来实现。

[0009] 本发明解决其技术问题所采用的技术方案是：

一种修改 BIOS SETUP 界面的方法，计算机系统在 UEFI Shell 环境中，所述方法使用的系统架构包括提供图形接口的图形抽象层（英文全称为 graphics abstraction layer，简称 GAL）、提供鼠标和键盘驱动的输入抽象层（英文全称为 input abstraction layer、简称 IAL）、图形设备接口层（英文全称为 graphics device interface、简称 GDI）、图形用户接口层（英文全称为 graphics user interface、简称 GUI）、应用程序编程接口（英文全称为 Application Programming Interface、简称 API），图形抽象层和输入抽象层位于系统最底层，图形设备接口层位于系统的中间层，图形用户接口层位于图形设备接口层上方，系统的最顶层为应用程序编程接口；

与图形抽象层和输入抽象层相对应的接口部分是 UEFI 中的 UGA 协议、Simple Pointer 协议和 Simple TextIn 协议，其中系统的图形抽象层对 UGA 协议进行封装，并为上层的图形设备接口层提供统一的接口，图形设备接口层则对图形抽象层进行封装，实现了更多的功能，如位图读取与绘制、区域剪切、文字输出等，为上层的图形用户接口层提供功能使用；输入抽象层则对 Simple Pointer 协议 与 SimpleTextIn 协议进行封装，并通过对鼠标键盘输入的监视生成消息 (Message)，通过消息可以进行窗口之间的通信，为图形用户接口层的（窗口创建与绘制、多窗口之间的剪切等）的功能提供服务；图形用户接口层通过应用程序编程接口访问 BIOS Setup 的应用程序；

所述方法步骤为：

（1）、在 UEFI 的 BIOS 源代码中添加图形抽象层和输入抽象层的驱动并实现相关的 UGA 协议、Simple Pointer 协议和 Simple TextIn 协议；

（2）、编译 UEFI 的 BIOS 源代码产生可执行的 ROM 文件；

（3）、更新步骤（2）中编译的 BIOS 源代码文件到计算机系统 Firmware 中；

（4）、编写 UEFI 应用程序实现 BIOS Setup 界面的显示和控制；

（5）、启动计算机系统配置文件到 UEFI Shell 模式下，运行应用程序。

[0010] 步骤（1）的实现，将图形抽象层和输入抽象层的程序按照驱动协议模型编写；包括如下步骤：①、编写 UGA 协议、Simple Pointer 协议和 Simple TextIn 协议的接口；②、用驱动格式实现上述接口。

[0011] 步骤（1）的①编写协议部分，需要在 BIOS 的 Protocol 文件夹下建立图形抽象层或输入抽象层的文件夹，并在该文件夹中编写 Protocol 的接口函数；

步骤（1）的②驱动格式是在 BIOS 的 Driver 文件夹下建立图形抽象层或输入抽象层的文件夹，在这个文件夹下编写 Protocol 文件中 GAL 函数接口的具体实现；需要实现 Supported 函数、Start 函数 与 Stop 函数；

Supported 函数实现步骤：用 UEFI 规范的 OpenProtocol 函数打开 UEFI 的 UGA 协议并加以测试，以保证之后的 GAL 功能函数可以调用 UGA 的服务，之后用 UEFI 规范的 CloseProtocol 函数关闭 UGA 协议；

Start 函数实现步骤：在得到 Supported 函数的保证下，用 OpenProtocol 函数打开 UEFI 的 UGA 协议以调用其提供的服务；随后构造相应的数据结构并用之前 Protocol 中的接口函数指针指向驱动程序中对应的函数；最后用 UEFI 规范的 InstallProtocolInterface 函数将 GAL 协议安装到 Controller Handle 上；

Stop 函数实现步骤：将 Start 中安装的协议用 CloseProtocol 关闭，并用 EFI 规范的 UninstallProtocolInterface 函数卸载，最后释放所构造的数据结构；

具备了上面 3 个函数的驱动程序就实现了 UEFI 驱动的绑定协议，为 GAL 功能函数的开发提供了一个符合 UEFI 驱动模式的框架；

接下来用图形学的算法实现相应的函数接口；

最后，通过在 UEFI 应用程序中调用 GAL 和 IAL 的驱动来实现对 BIOS Setup 界面的显示和修改；这包括图形设备接口的实现、字符的显示、窗口的管理和控制操作；窗口的管理和控制通过消息传递的方式来实现，通过调用 IAL 的驱动产生消息传递给窗口控制程序；

对于 BIOS Setup 菜单的显示需要调用 UEFI 运行时服务的接口函数 GetVariable 和 GetNextVariableName 以及 QueryVariableInfo；在对 Setup 菜单的数值进行修改后需要调用 SetVariable 接口函数。具体的实现可以参考目前 UEFI BIOS 的实现方法。

[0012] 本发明的一种修改 BIOS SETUP 界面的方法和现有技术相比，基于 UEFI 系统中 GAL 与 IAL 的分层设计大大降低了系统的耦合度，提高了系统的可扩展性。GAL 与 IAL 和 UEFI 驱动模型的集成，为系统在 UEFI SHELL 下实现修改 BIOS SETUP 界面提供了坚实的基础，使系统取得了较好的交互体验和较美观的界面效果；因而，具有很好的推广使用价值。

附图说明

[0013] 下面结合附图对本发明进一步说明。

[0014] 附图 1 为一种修改 BIOS SETUP 界面的方法使用的系统架构的结构框图。

具体实施方式

[0015] 下面结合附图和具体实施例对本发明作进一步说明。

[0016] 实施例 1：

一种修改 BIOS SETUP 界面的方法，计算机系统在 UEFI Shell 环境中，所述方法使用的系统架构包括提供图形接口的图形抽象层（英文全称为 graphics abstraction layer，简称 GAL）、提供鼠标和键盘驱动的输入抽象层（英文全称为 input abstraction layer、简称 IAL）、图形设备接口层（英文全称为 graphics device interface、简称 GDI）、图形用户接口层（英文全称为 graphics user interface、简称 GUI）、应用程序编程接口（英文全称为 Application Programming Interface、简称 API），图形抽象层和输入抽象层位于系统最底层，图形设备接口层位于系统的中间层，图形用户接口层位于图形设备接口层上方，系统的最顶层为应用程序编程接口；

与图形抽象层和输入抽象层相对应的接口部分是 UEFI 中的 UGA 协议、Simple Pointer 协议和 Simple TextIn 协议，其中系统的图形抽象层对 UGA 协议进行封装，并为上层的图形设备接口层提供统一的接口，图形设备接口层则对图形抽象层进行封装，实现了更多的功能，如位图读取与绘制、区域剪切、文字输出等，为上层的图形用户接口层提供功能使用；输入抽象层则对 Simple Pointer 协议 与 SimpleTextIn 协议进行封装，并通过对鼠标键盘输入的监视生成消息 (Message)，通过消息可以进行窗口之间的通信，为图形用户接口层的（窗口创建与绘制、多窗口之间的剪切等）的功能提供服务；图形用户接口层通过应用程序编程接口访问 BIOS Setup 的应用程序；

所述方法步骤为：

(1)、在UEFI的BIOS源代码中添加图形抽象层和输入抽象层的驱动并实现相关的UGA协议、Simple Pointer 协议和Simple TextIn协议；

(2)、编译UEFI的BIOS源代码产生可执行的ROM文件；

(3)、更新步骤(2)中编译的BIOS源代码文件到计算机系统Firmware中；

(4)、编写UEFI应用程序实现BIOS Setup界面的显示和控制；

(5)、启动计算机系统配置文件到UEFI Shell模式下,运行应用程序。

[0017] 实施例2：

一种修改BIOS SETUP界面的方法,计算机系统在UEFI Shell环境中,所述方法使用的系统架构包括提供图形接口的图形抽象层(英文全称为graphics abstraction layer,简称GAL)、提供鼠标和键盘驱动的输入抽象层(英文全称为input abstraction layer、简称IAL)、图形设备接口层(英文全称为graphics device interface、简称GDI)、图形用户接口层(英文全称为graphics user interface、简称GUI)、应用程序编程接口(英文全称为Application Programming Interface、简称API),图形抽象层和输入抽象层位于系统最底层,图形设备接口层位于系统的中间层,图形用户接口层位于图形设备接口层上方,系统的最顶层为应用程序编程接口；

与图形抽象层和输入抽象层相对应的接口部分是UEFI中的UGA协议、Simple Pointer协议和Simple TextIn协议,其中系统的图形抽象层对UGA 协议进行封装,并为上层的图形设备接口层提供统一的接口,图形设备接口层则对图形抽象层进行封装,实现了更多的功能,如位图读取与绘制、区域剪切、文字输出等,为上层的图形用户接口层提供功能使用；输入抽象层则对Simple Pointer协议 与SimpleTextIn协议进行封装,并通过对鼠标键盘输入的监视生成消息(Message),通过消息可以进行窗口之间的通信,为图形用户接口层的(窗口创建与绘制、多窗口之间的剪切等)的功能提供服务;图形用户接口层通过应用程序编程接口访问BIOS Setup的应用程序；

所述方法步骤为：

(1)、在UEFI的BIOS源代码中添加图形抽象层和输入抽象层的驱动并实现相关的UGA协议、Simple Pointer 协议和Simple TextIn协议；

(2)、编译UEFI的BIOS源代码产生可执行的ROM文件；

(3)、更新步骤(2)中编译的BIOS源代码文件到计算机系统Firmware中；

(4)、编写UEFI应用程序实现BIOS Setup界面的显示和控制；

(5)、启动计算机系统配置文件到UEFI Shell模式下,运行应用程序。

[0018] 步骤(1)的实现,将图形抽象层和输入抽象层的程序按照驱动协议模型编写;包括如下步骤:①、编写UGA协议、Simple Pointer 协议和Simple TextIn协议的接口;②、用驱动格式实现上述接口。

[0019] 实施例3：

一种修改BIOS SETUP界面的方法,计算机系统在UEFI Shell环境中,所述方法使用的系统架构包括提供图形接口的图形抽象层(英文全称为graphics abstraction layer,简称GAL)、提供鼠标和键盘驱动的输入抽象层(英文全称为input abstraction layer、简称IAL)、图形设备接口层(英文全称为graphics device interface、简称GDI)、图形用户

接口层(英文全称为graphics user interface、简称GUI)、应用程序编程接口(英文全称为Application Programming Interface、简称API),图形抽象层和输入抽象层位于系统最底层,图形设备接口层位于系统的中间层,图形用户接口层位于图形设备接口层上方,系统的最顶层为应用程序编程接口;

与图形抽象层和输入抽象层相对应的接口部分是UEFI中的UGA协议、Simple Pointer协议和Simple TextIn协议,其中系统的图形抽象层对UGA协议进行封装,并为上层的图形设备接口层提供统一的接口,图形设备接口层则对图形抽象层进行封装,实现了更多的功能,如位图读取与绘制、区域剪切、文字输出等,为上层的图形用户接口层提供功能使用;输入抽象层则对Simple Pointer协议与SimpleTextIn协议进行封装,并通过对鼠标键盘输入的监视生成消息(Message),通过消息可以进行窗口之间的通信,为图形用户接口层的(窗口创建与绘制、多窗口之间的剪切等)的功能提供服务;图形用户接口层通过应用程序编程接口访问BIOS Setup的应用程序;

所述方法步骤为:

(1)、在UEFI的BIOS源代码中添加图形抽象层和输入抽象层的驱动并实现相关的UGA协议、Simple Pointer协议和Simple TextIn协议;

(2)、编译UEFI的BIOS源代码产生可执行的ROM文件;

(3)、更新步骤(2)中编译的BIOS源代码文件到计算机系统Firmware中;

(4)、编写UEFI应用程序实现BIOS Setup界面的显示和控制;

(5)、启动计算机系统配置文件到UEFI Shell模式下,运行应用程序。

[0020] 步骤(1)的实现,将图形抽象层和输入抽象层的程序按照驱动协议模型编写;包括如下步骤:①、编写UGA协议、Simple Pointer协议和Simple TextIn协议的接口;②、用驱动格式实现上述接口。

[0021] 步骤(1)的①编写协议部分,需要在BIOS的Protocol文件夹下建立图形抽象层或输入抽象层的文件夹,并在该文件夹中编写Protocol的接口函数;

步骤(1)的②驱动格式是在BIOS的Driver文件夹下建立图形抽象层或输入抽象层的文件夹,在这个文件夹下编写Protocol文件中GAL函数接口的具体实现;需要实现Supported函数、Start函数与Stop函数;

Supported函数实现步骤:用UEFI规范的OpenProtocol函数打开UEFI的UGA协议并加以测试,以保证之后的GAL功能函数可以调用UGA的服务,之后用UEFI规范的CloseProtocol函数关闭UGA协议;

Start函数实现步骤:在得到Supported函数的保证下,用OpenProtocol函数打开UEFI的UGA协议以调用其提供的服务;随后构造相应的数据结构并用之前Protocol中的接口函数指针指向驱动程序中对应的函数;最后用UEFI规范的InstallProtocolInterface函数将GAL协议安装到Controller Handle上;

Stop函数实现步骤:将Start中安装的协议用CloseProtocol关闭,并用EFI规范的UninstallProtocolInterface函数卸载,最后释放所构造的数据结构;

具备了上面3个函数的驱动程序就实现了UEFI驱动的绑定协议,为GAL功能函数的开发提供了一个符合UEFI驱动模式的框架;

接下来用图形学的算法实现相应的函数接口;

最后，通过在UEFI应用程序中调用GAL和IAL的驱动来实现对BIOS Setup界面的显示和修改；这包括图形设备接口的实现、字符的显示、窗口的管理和控制操作；窗口的管理和控制通过消息传递的方式来实现，通过调用IAL的驱动产生消息传递给窗口控制程序；

对于BIOS SETUP菜单的显示需要调用UEFI运行时服务的接口函数GetVariable和GetNextVariableName以及QueryVariableInfo；在对SETUP菜单的数值进行修改后需要调用SetVariable接口函数。具体的实现可以参考目前UEFI BIOS的实现方法。

[0022] 上述具体实施方式仅是本发明的具体个案，本发明的专利保护范围包括但不限于上述具体实施方式，任何符合本发明的一种修改BIOS SETUP界面的方法的权利要求书的且任何所属技术领域的普通技术人员对其所做的适当变化或替换，皆应落入本发明的专利保护范围。

[0023] 除说明书所述的技术特征外，均为本专业技术人员的已知技术。

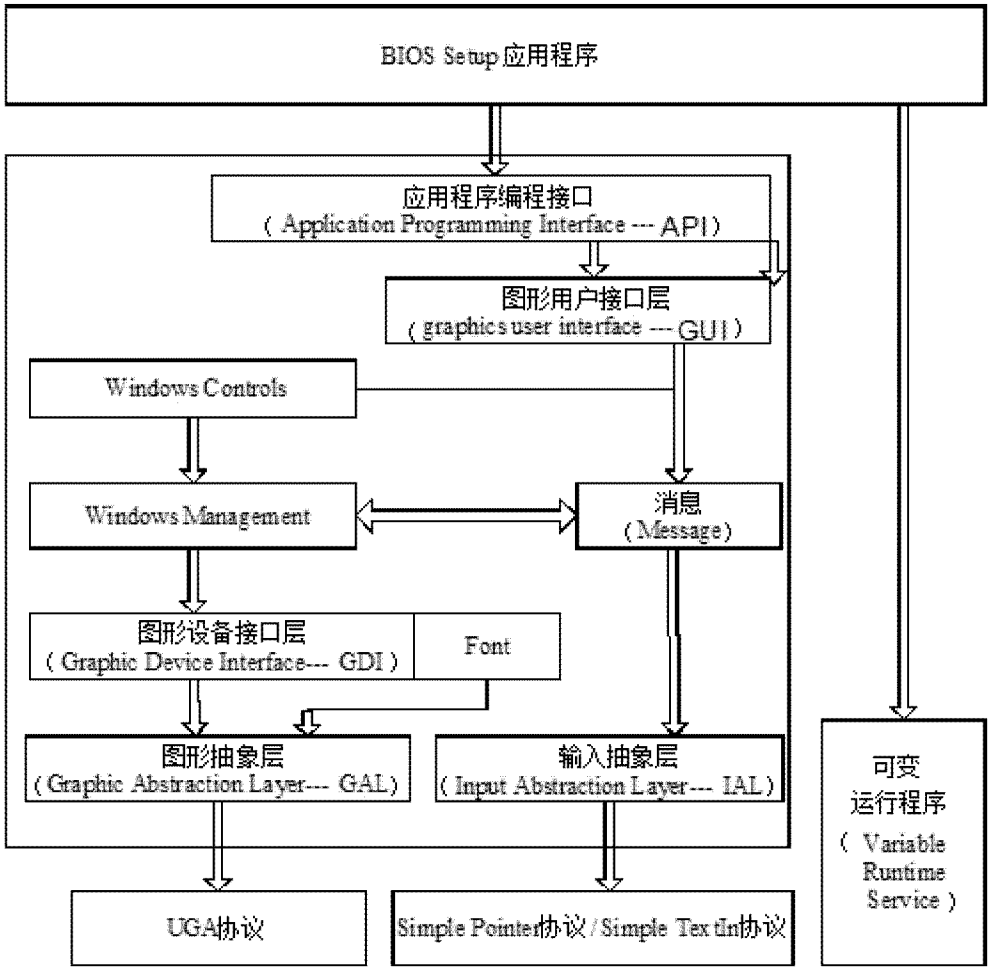

BIOS Setup应用程序
应用程序编程接口
(Application Programming Interface --- API)
图形用户接口层
(graphics user interface --- GUI)
Windows Controls
Windows Management
消息
(Message)
图形设备接口层
(Graphic Device Interface--- GDI)
Font
图形抽象层
(Graphic Abstraction Layer--- GAL)
输入抽象层
(Input Abstraction Layer--- IAL)
UGA协议
Simple Pointer协议/Simple Textin协议
可变
运行程序
(Variable
Runtime
Service)

图 1