

[0006] Fig. 3 is a block diagram of another system supporting on-demand compiling and deployment of mobile applications;

[0007] Fig. 4 is a block diagram of a representative programmable device; and

5 [0008] Fig. 5 is a block diagram of an exemplary packet structure. ~~for conveying customized applications; and~~

[0009] Fig. 6 is a flow chart illustrating a method of creating customized applications for programmable devices.

#### DETAILED DESCRIPTION

10 [0010] Although the following text sets forth a detailed description of numerous different embodiments, it should be understood that the legal scope of the description is defined by the words of the claims set forth at the end of this disclosure. The detailed description is to be construed as exemplary only and does not describe every possible embodiment since describing every possible embodiment would be impractical, if not impossible. Numerous alternative embodiments could be implemented, using either current technology or  
15 technology developed after the filing date of this patent, which would still fall within the scope of the claims.

[0011] It should also be understood that, unless a term is expressly defined in this patent using the sentence “As used herein, the term ‘ \_\_\_\_\_ ’ is hereby defined to mean...” or a similar sentence, there is no intent to limit the meaning of that term, either expressly or by  
20 implication, beyond its plain or ordinary meaning, and such term should not be interpreted to be limited in scope based on any statement made in any section of this patent (other than the language of the claims). To the extent that any term recited in the claims at the end of this patent is referred to in this patent in a manner consistent with a single meaning, that is done for sake of clarity only so as to not confuse the reader, and it is not intended  
25 that such claim term be limited, by implication or otherwise, to that single meaning.

Finally, unless a claim element is defined by reciting the word “means” and a function without the recital of any structure, it is not intended that the scope of any claim element be interpreted based on the application of 35 U.S.C. § 112, sixth paragraph.

30 [0012] Much of the inventive functionality and many of the inventive principles are best implemented with or in software programs or instructions and integrated circuits (ICs) such as application specific ICs. It is expected that one of ordinary skill, notwithstanding possibly significant effort and many design choices motivated by, for example, available time, current technology, and economic considerations, when guided by the concepts and principles disclosed herein will be readily capable of generating such software instructions

and programs and ICs with minimal experimentation. Therefore, in the interest of brevity and minimization of any risk of obscuring the principles and concepts in accordance to the present invention, further discussion of such software and ICs, if any, will be limited to the essentials with respect to the principles and concepts of the preferred embodiments.

5 **[0013]** With reference to Fig. 1, an exemplary system for implementing the claimed method and apparatus includes a general purpose computing device in the form of a computer 110. While the programmable device described in more detail below may typically be a handheld or portable computing/communication device, it may be a computer as illustrated here. The computer 110 may also represent a typical server as  
10 described below for use in fulfilling requests from a programmable device. Components shown in dashed outline are not technically part of the computer 110, but are used to illustrate the exemplary embodiment of Fig. 1. Components of computer 110 may include, but are not limited to, a processor 120, a system memory 130, a memory/graphics interface 121, also known as a Northbridge chip, and an I/O interface 122, also known as a  
15 Southbridge chip. The system memory 130 and a graphics processor 190 may be coupled to the memory/graphics interface 121. A monitor 191 or other graphic output device may be coupled to the graphics processor 190.

**[0014]** A series of system busses may couple various system components including a high speed system bus 123 between the processor 120, the memory/graphics interface 121 and  
20 the I/O interface 122, a front-side bus 124 between the memory/graphics interface 121 and the system memory 130, and an advanced graphics processing (AGP) bus 125 between the memory/graphics interface 121 and the graphics processor 190. The system bus 123 may be any of several types of bus structures including, by way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel  
25 Architecture (MCA) bus and Enhanced ISA (EISA) bus. As system architectures evolve, other bus architectures and chip sets may be used but often generally follow this pattern. For example, companies such as Intel and AMD support the Intel Hub Architecture (IHA) and the Hypertransport™ architecture, respectively.

**[0015]** The computer 110 typically includes a variety of computer readable media.

30 Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any

method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes,  
5 magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110.

**[0016]** The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access  
10 memory (RAM) 132. The system ROM 131 may contain permanent system data 143, such as identifying and manufacturing information. In some embodiments, a basic input/output system (BIOS) may also be stored in system ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processor 120. By way of example, and not limitation, Fig. 1  
15 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

**[0017]** The I/O interface 122 may couple the system bus 123 with a number of other busses 126, 127 and 128 that couple a variety of internal and external devices to the computer 110. A serial peripheral interface (SPI) bus 126 may connect to a basic  
20 input/output system (BIOS) memory 133 containing the basic routines that help to transfer information between elements within computer 110, such as during start-up.

**[0018]** A super input/output chip 160 may be used to connect to a number of ‘legacy’ peripherals, such as floppy disk 152, keyboard/mouse 162, and printer 196, as examples. The super I/O chip 160 may be connected to the I/O interface 122 with a bus 127, such as  
25 a low pin count (LPC) bus, in some embodiments. Various embodiments of the super I/O chip 160 are widely available in the commercial marketplace.

**[0019]** In one embodiment, bus 128 may be a Peripheral Component Interconnect (PCI) bus, or a variation thereof, may be used to connect higher speed peripherals to the I/O interface 122. A PCI bus may also be known as a Mezzanine bus. Variations of the PCI  
30 bus include the Peripheral Component Interconnect-Express (PCI-E) and the Peripheral Component Interconnect – Extended (PCI-X) busses, the former having a serial interface and the latter being a backward compatible parallel interface. In other embodiments, bus 128 may be an advanced technology attachment (ATA) bus, in the form of a serial ATA bus (SATA) or parallel ATA (PATA).

We claim: CLAIMS

1. A method ~~(600) of creating applications responsive to a request from a programmable device (300)~~ comprising:
  - ~~determining a base configuration (602) of the programmable device (300);~~
  - 5 ~~presenting (606) a user interface (310) with a list of available applications compatible with the base configuration of the programmable device;~~
  - ~~receiving (608) user input relating to a selected application from the list of available applications for installation on the programmable device;~~
  - ~~collecting (610) a specific configuration corresponding to the programmable~~
  - 10 ~~device (300);~~
  - ~~constructing (612) a request package at the programmable device, the request package including component elements of the specific configuration including at least an identifier of the selected application, a script engine version identifier, and at least one of a device identifier and a device type identifier;~~
  - 15 ~~sending (614) the request package from the programmable device to a fulfillment center;~~
  - ~~receiving (616) the a request package at the fulfillment center;~~ a computing device acting as a server, wherein the request package is received from a programmable device and includes component elements of a specific configuration of the programmable device,
  - 20 the component elements including at least:
    - an identifier of a selected application for installation on the programmable device,
    - a script engine version identifier identifying a version of a script engine on the programmable device, wherein the version of the script engine is configured to execute
    - multiple different applications including the selected application, and
    - 25 at least one of a device identifier of the programmable device or a device type identifier of the programmable device;
  - ~~parsing (618) the request package with the specific configuration of the programmable device~~ into the component elements of the request package;
  - ~~creating (622) a custom version of the selected application~~ based on the
  - 30 ~~corresponding to the~~ component elements of the request package, wherein the custom version of the selected application is executable by the version of the script engine on the programmable device; and
  - ~~sending (624) the custom version of the selected application~~ from the computing device acting as the server to the programmable device.

2. The method of claim 1, further comprising sending ~~(604) the a~~ list of multiple available applications compatible with ~~a~~the base configuration of the programmable device from the computing device acting as the server to the programmable device.

3. The method of claim 12, further comprising:

~~5 sending at least a portion of the request package from the fulfillment center to a back-end compiler;~~

creating ~~(620)~~ a custom compilation configuration according to the component elements of the specific configuration of the programmable device; and

creating~~compiling~~ the custom version of the~~selected~~ application after receiving the  
10 request package by compiling the selected application using the custom compilation configuration~~resulting in the custom version of the selected application targeted to the programmable device.~~

4. The method of claim 3, wherein:

creating the custom compilation configuration comprises creating a custom  
15 makefile with specific compilation options corresponding to the specific configuration of the programmable device~~;~~, and

~~4.~~ creating the custom version of the selected application comprises compiling the application using the custom makefile.

5. The method of claim 13, further comprising:

~~20 delivering (N.B.i.e., confirming delivery as opposed to sending.624) the custom version of the selected application to the programmable device; and~~

deleting all copies of the custom version of the selected application from the computing device acting as the server~~back-end compiler and the fulfillment center~~ following confirming delivery of the custom version of the selected application.

25 6. The method of claim 1, wherein the component elements of the request package identify one or more of~~creating the custom version of the selected application comprises:~~

~~pre-selecting component elements of the request package;~~

~~creating the custom version of the selected application in accordance with the pre-selected component elements; and~~

~~30 storing the custom version of the selected application and pre-selected component elements.~~

an external light sensor of the programmable device;

a motion sensor capability of the programmable device;

a camera type of the programmable device;

a sound output capability of the programmable device;  
a peripheral configuration of the programmable device;  
a total and an available non-volatile memory of the programmable device; or  
a total volatile memory of the programmable device.

5 ~~7. — The method of claim 6, wherein sending the custom version of the selected application to the programmable device comprises:~~

~~matching component elements of the request package with stored anticipated component elements; and~~

~~10 sending to the programmable device the custom version of the selected application corresponding with pre-selected component elements matching the component elements of the request package.~~

~~8. — The method of claim 1, wherein determining the base configuration of the programmable device comprises:~~

~~15 determining a type and version of an operating system running on the programmable device;~~

~~determining a screen size in pixels and a color depth;~~

~~determining an input device type.~~

~~9. — The method of claim 1, wherein collecting the specific configuration corresponding to the programmable device comprises:~~

~~20 determining an external light sensor;~~

~~determining a motion sensor capability;~~

~~determining a camera type;~~

~~determining a sound output capability;~~

~~determining a peripheral configuration;~~

~~25 determining a total and an available non-volatile memory; and~~

~~determining a total volatile memory.~~

~~10. — The method of claim 1, further comprising:~~

~~30 examining a customization header of the custom version of the selected application, the customization header containing selected information corresponding to the component elements used in creating the custom version of the selected application; and~~

~~determining that the custom version of the selected application matches the specific configuration corresponding to the programmable device.~~

11. ~~—The method of claim 11, further comprising deferring installation when the custom version of the selected application does not match the specific configuration corresponding to the programmable device.~~

12. ~~—The method of claim 11, further comprising installing the custom version of the selected application when the custom version of the selected application matches the specific configuration.~~

13. ~~—The method of claim 1, wherein collecting the specific configuration corresponding to the programmable device comprises:~~

~~extracting a catalog of desired specific configuration data from the list of available application; and~~

~~collecting the desired specific configuration data according to the catalog.~~

14. ~~—The method of claim 1, further comprising:~~

~~rejecting a request package when the programmable device is not in a database of known programmable devices; and~~

~~requesting additional configuration information from the programmable device.~~

15. 7. A programmable device ~~supporting download of applications~~ comprising:

a processing unit ~~(302)~~;

a transceiver ~~(304)~~ coupled to the processing unit and configured to supporting two-way communications ~~coupled to the processing unit~~;

a memory ~~(312)~~ coupled to the processing unit and storing instructions configured to execute ~~for execution~~ on the processing unit, the instructions configured to implementing an operating system ~~(320)~~, a download system ~~(320)~~, and a script engine ~~(316)~~; and

a user interface ~~(308)(310)~~, coupled to the processing unit and supported by the operating system,

wherein the script engine is configured to execute multiple different applications including a custom version of a selected application, and

wherein the download system is configured to:

receives a list of qualified applications ~~appropriate to the programmable device~~ via the transceiver,

causes the list to be displayed on the user interface,

receives input from a user related to ~~the~~ a selected application,

collects specific configuration information corresponding to a current configuration of ~~about~~ the programmable device, the specific configuration information

including an identifier of a version of the script engine associated with the current configuration.

sends a request for the selected application to a server, with the request sent to the server comprising the specific configuration information including the identifier of the version of the script engine associated with the current configuration of the programmable device, and

receives, from the server, ~~the~~ a customized version of the selected application, ~~appropriate to the specific configuration information,~~ wherein the custom version of the selected application received from the server is executable by the version of the script engine that is associated with the current configuration of the programmable device.

8. The programmable device of claim 7, wherein the download system is further configured to:

examine a customization header of the custom version of the selected application, the customization header containing selected information corresponding to component elements of the specific configuration information used in creating the custom version of the selected application; and

determine whether the customization header matches the current configuration of the programmable device.

9. The programmable device of claim 8, wherein the download system is further configured to:

defer installation of the custom version of the selected application when the customization header does not match the current configuration of the programmable device.

10. The programmable device of claim 8, wherein the download system is further configured to:

install the custom version of the selected application on the programmable device when the customization header matches the current configuration of the programmable device.

11. The programmable device of claim 7, wherein the download system is further configured to:

collect the specific configuration information according to a catalog that identifies desired specific configuration data for the selected application.

12. The programmable device of claim 7, wherein the script engine is further configured to:

interpret the selected application.

13. A server comprising:

a processing unit; and

computer readable instructions which, when executed by the processing unit,

5 configure the processing unit to:

receive, at the server, a request from a programmable device, the request including  
a set of component elements comprising:

an identifier of a selected application,

a script engine version identifier, and

10 at least one of a device identifier of the programmable device or a device type  
identifier of the programmable device,

wherein the script engine version identifier identifies a version of the script engine  
on the programmable device and the version of the script engine is configured to execute  
multiple different applications including the selected application;

15 evaluate individual component elements of the set to identify available options for  
the selected application;

select one or more option settings for the available options according to the  
individual component elements;

create a custom version of the selected application using the one or more option  
20 settings, wherein the custom version of the selected application is created using at least an  
individual option setting that is selected based on the version of the script engine on the  
programmable device and the custom version of the selected application is executable by  
the version of the script engine on the programmable device; and

send the custom version of the selected application from the server to the  
25 programmable device.

14. The server of claim 13, wherein the computer readable instructions, when  
executed, further configure the processing unit to:

determine a standard configuration of the programmable device;

develop a list of multiple applications that are supported by the standard  
30 configuration of the programmable device; and

send the list of multiple applications to the programmable device, wherein the  
selected application is selected from the list.

15. The server of claim 13, wherein the computer readable instructions further  
configure the processing unit to:

create the custom version of the selected application using one or more other individual option settings that correspond to:

a screen size and color depth of the programmable device;

a motion sensor capability of the programmable device;

5 a camera type of the programmable device;

a sound output capability of the programmable device;

a wireless peripheral support capability of the programmable device;

a playback support of the programmable device; or

a memory configuration of the programmable device.

10 16. The server of claim 13, wherein the computer readable instructions further configure the processing unit to:

evaluate a network type to select the one or more option settings.

17. The server of claim 13, wherein the one or more option settings comprise compiler options or custom include files.

15 18. The server according to claim 13, wherein the one or more option settings include one or more add-on features in the custom version of the selected application or exclude the one or more add-on features from the custom version of the application.

19. The server according to claim 13, wherein the computer readable instructions further configure the processing unit to:

20 pre-compile another version of the selected application for a common device configuration; and

send the pre-compiled another version of the selected application to another programmable device having the common device configuration.

25 20. The server according to claim 13, wherein the custom version of the selected application is an applet or executable script.

21. The server according to claim 13, embodied as a general purpose computing device.