

The opinion in support of the decision being entered today was **not** written for publication and is **not** binding precedent of the Board.

Paper No. 24

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

---

**Ex parte** DEAN YU and CHRIS DEROSI

---

Appeal No. 1999-0080  
Application No. 08/558,929

---

ON BRIEF

---

Before DIXON, GROSS, and LEVY, **Administrative Patent Judges**.  
DIXON, **Administrative Patent Judge**.

**DECISION ON APPEAL**

This is a decision on appeal from the examiner's final rejection of claims 2-6, 11-17 and 21-27, which are all of the claims pending in this application.

We REVERSE.

**BACKGROUND**

The appellants' invention relates to a method and apparatus for enabling a computer system. The invention uses enabler files/programs to adapt the operating

system to later developed hardware and system changes and store the updated operating system. An understanding of the invention can be derived from a reading of exemplary claim 21, which is reproduced below.

21. A computing system comprising:

one of various types of processors for executing software; and

a software operating system for use by said processor, the operating system comprising

a boot-up file for beginning execution of an initial portion of a boot-up routine which initial portion of said boot-up routine identifies the type of processor present and passes execution of the boot-up routine; and

a self-contained enabler file, containing processor-specific information, which receives execution of the boot-up routine from said operating system and enables said operating system to execute application programs in the identified one of various types of processors using said processor-specific information, said enabler file being initially stored in a read-write memory device so that said enabler file may be replaced with an updated enabler file when system changes are made in said computing system.

The prior art references of record relied upon by the examiner in rejecting the appealed claims are:

Schmidt et al. (Schmidt)	4,558,413	Dec. 10, 1985
Mitani et al. (Mitani)	4,620,273	Oct. 28, 1986
Arnold et al. (Arnold)	5,128,995	Jul. 07, 1992
Sherer et al. (Sherer)	5,459,854	Oct. 17, 1995 (Filed Jul. 09, 1991)

Appeal No. 1999-0080  
Application No. 08/558,929

Claims 2, 3, 5, 6, 11, 12, 14, 15, and 21-26 stand rejected under 35 U.S.C. § 103 as being unpatentable over Sherer in view of Arnold. Claims 16 and 27 stand rejected under 35 U.S.C. § 103 as being unpatentable over Sherer and Arnold in view of Schmidt. Claims 4 and 13 stand rejected under 35 U.S.C. § 103 as being unpatentable over Sherer and Arnold in view of Mitani.

Rather than reiterate the conflicting viewpoints advanced by the examiner and appellants regarding the above-noted rejections, we make reference to the examiner's answer (Paper No. 23, mailed Feb. 27, 1998) and the examiner's action (Paper No. 15, mailed Sep. 16, 1996) for the examiner's reasoning in support of the rejections, and to appellants' brief (Paper No. 22, filed Dec. 8, 1997) for appellants' arguments thereagainst.

### **OPINION**

In reaching our decision in this appeal, we have given careful consideration to appellants' specification and claims, to the applied prior art references, and to the respective positions articulated by appellants and the examiner. As a consequence of our review, we make the determinations which follow.

As evidence of obviousness the examiner relies on the teachings of Sherer in combination with the teachings of Arnold to suggest the obviousness of the invention as recited in independent claims 21-26.

Appeal No. 1999-0080  
Application No. 08/558,929

In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness. **See In re Rijckaert**, 9 F.3d 1531, 1532, 28 USPQ2d 1955, 1956 (Fed. Cir. 1993). A *prima facie* case of obviousness is established by presenting evidence that the reference teachings would appear to be sufficient for one of ordinary skill in the relevant art having the references before him to make the proposed combination or other modification. **See In re Lintner**, 9 F.2d 1013, 1016, 173 USPQ 560, 562 (CCPA 1972). Furthermore, the conclusion that the claimed subject matter is *prima facie* obvious must be supported by evidence, as shown by some objective teaching in the prior art or by knowledge generally available to one of ordinary skill in the art that would have led that individual to combine the relevant teachings of the references to arrive at the claimed invention. **See In re Fine**, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

The examiner states that “[i]t would have been obvious to one of ordinary skill in the art to execute the boot-up routine and then the enabler file so that system compatibility can be determined before loading lengthy operating system code” (Office action at page 4 as incorporated by the examiner at page 5 of the answer), but the examiner provides no support or motivation in the prior art applied for this conclusion.

The portion of Sherer relied upon by the examiner is columns 4 and 5, which state in relevant part

[i]ndividual elements 16, 18, 20, 22 represent different encoded operating system segments forming a part of the operating system program, but each optimized for a specific microprocessor 30 (of a type yet to be identified), and program element 24 represents an encoded operating system segment which operates equally well with all of the types of microprocessor 30. Included is a program element 26 which is useful to identify the type of microprocessor. This program element 26 may be a set of instructions which executes uniquely with each of the types of microprocessors and which returns a flag of some value which identifies the type of microprocessor. It may be executable code comprising a series of tests which selectively eliminates microprocessor types as choices without resulting in an abortive attempt at execution of instructions.

The process according to the invention starts by loading all of the program elements 16, 18, 20, 22, 24, 26 of the operating system from a mass storage system 14 into system memory 28 (Step A, FIG. 1).

The program element 26 is then executed by the microprocessor 30 as the test for the type of the microprocessor 30 in use (Step B). With each test, a flag signal 32 is returned to the program module as a parameter for use in further execution of the program element 26 until the type of microprocessor in use is uniquely identified.

Once the type of microprocessor is identified, the program element 26 then identifies which portions 34 of itself are still needed and which portions 36, 38, of itself are no longer needed (FIG. 3), and the memory locations corresponding to those unneeded portions 36, 38 are released or freed for use by other programs (Step C). In order to compact the storage of the needed portion 34, the needed portion 34 may be relocated within the memory 28 (Step D and FIG. 4). Pointers to the needed portions 34 are reset (Step E), and other portions of the operating system (if any) can be loaded (Step F).

The relocated portions 16, 18, 10, or 22 may be within user-accessible memory space, or in protected memory space as portions 16', 18', 20', or 22', as shown in FIG. 4. The choice is typically microprocessor dependent. Control of the computer can then be turned over to other processes.

Here, Sherer discloses that the unnecessary procedures associated with various versions of drivers are overwritten or discarded after initialization. (See Col. 5.)

The examiner relies on column 6 to teach the initialization code for beginning the initial portion of a boot up routine. But in column 6, Sherer is concerned with a device driver rather than the boot up at the very beginning of initialization.

The relevant portion of columns 6 and 7 are reproduced below:

[a]s mentioned above, when the device driver is first loaded in the memory, each of the performance critical program segments is composed of at least one code block, each code block is optimized for one or more particular variant architectures, within which the device driver is intended to run.

The initialization process for this preferred embodiment is illustrated in FIG. 7. It begins when the operating system reads the program image of the device driver from a secondary storage device and loads it into memory. The operating system then branches to the initialization code of the program, InitCodeSeg, which has been loaded into memory.

The initialization process includes the nine steps illustrated in FIG. 7.

As mentioned above, the process begins by loading the program and calling the initialization code (block 70). The first step involves printing a message on the display terminal identifying the software which is being

initialized (block 71). Next, the device driver searches for an unused device driver name among all other device drivers which have been loaded (block 72). In the third step, the protocol manager is opened to obtain a pointer to a configuration memory image, if such configuration memory exists (block 73). Next, the configuration memory image is parsed to identify key words and parameters that inform the device driver of the host architecture (block 74).

In step 5, the expansion slots of the host computer are scanned for an adapter which will be controlled by this device driver. Once found, the adapter's configuration information is read (block 75). In step 6, the variables which could not be initialized at compile time are now initialized and code is executed which determines the device driver's host environment (block 76).

In step 7, the code blocks are selected and relocated, possibly by overwriting code blocks which are not needed. The data structures which were initialized at compile time and in step 6 (block 76) are used to guide the relocation process. The host environment which was determined in steps 4 (block 74) and 6 (block 76) is also used to guide the selection and relocation process (block 77). In step 8, the device drivers timer interrupt service routine is registered so that it will be called during each timer tick (block 78). In step 9, the device driver informs the protocol manager that it has been initialized and is able to bind with other modules in the system (block 79). Finally, the control returns to the host operating system (block 80).

In our view, this is not "initialization code" (see Office action at page 3) as maintained by the examiner. Appellants argue that the examiner's position that a skilled artisan would have been motivated to load separate files rather than bigger files, is in error and based upon impermissible hindsight. (See brief at pages 4-5.) We agree with

appellants that the examiner has not shown support for the maintained position from the prior art applied or set forth a convincing line of reasoning for this conclusion. Therefore, we agree with appellants that the examiner has based his rejection upon hindsight. The examiner maintains that “Arnold checks compatibility and does not load [the] operating system if the hardware is not compatible. . . .” (See answer at page 6.) Here, the loading of the ROM-BIOS and then (power on self test (POST) or) master boot record (MBR) separation is for conservation of space in the ROM and not for the updating or modification of an enabler file as recited in claim 21.

The examiner relies upon the claim language “may be replaced” for a broad claim interpretation (answer at page 9-10) and suggests that “it is clear that when new architectures emerge, Sherer et al will replace the files with an updated version; e.g. elements 16, 18, 20, and 22 will have newer architectures.” The examiner provides no support for this finding. We disagree with the examiner that it is clear that there would be an update. We find that this is speculation and conjecture by the examiner. In our view, Sherer and Arnold do not teach or suggest the updating of the enabler file.

The examiner maintains that “Sherer et al and Arnold et al can in fact be updated anytime a change is made.” (See answer at page 11.) Here again, the examiner speculates about updating, but the examiner points to no teaching in the

reference to teach or suggest an enabler file which can be updated. The examiner further maintains that the “Sherer et al reference is designed to accommodate new hardware features . . . Sherer and Arnold store information in files, they can update programs for the variant architectures.” (See answer at page 12.) We disagree with the examiner. In our view, Sherer discloses a system for variant architectures and not necessarily for future/new hardware. Again, this appears to be speculation on the part of the examiner.

Appellants argue “unexpected results,” but do not provide any evidence beyond bare arguments. (See brief at pages 6-7.) Therefore, this argument is not persuasive.

Appellants argue that Sherer is directed to memory minimization and system efficiency rather than replacing an enabler routine. (See brief at page 8.) We agree with appellants. In our view, neither Sherer nor Arnold teach or suggest the “said enabler file being initially stored in a read-write memory device so that said enabler file may be replaced with an updated enabler file when system changes are made in said computing system” as recited in claim 21. Neither Sherer nor Arnold address that the enabler file “may be replaced with an updated enabler file when system changes are made in said computing system” as recited in the claim, and the examiner has not provided a motivation for the replacement with an updated enabler file.

When it is necessary to select elements of various teachings in order to form the claimed invention, we ascertain whether there is any suggestion or motivation in the prior art to make the selection made by the appellants. Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching, suggestion or incentive supporting the combination. The extent to which such suggestion must be explicit in, or may be fairly inferred from the references, is decided on the facts of each case, in light of the prior art and its relationship to the appellants' invention. As in all determinations under 35 U.S.C. § 103, the decision maker must bring judgment to bear. It is impermissible, however, simply to engage in a hindsight reconstruction of the claimed invention, using the appellants' structure as a template and selecting elements from references to fill the gaps. The references themselves must provide some teaching whereby the appellants' combination would have been obvious. **In re Gorman**, 933 F.2d 982, 986, 18 USPQ2d 1885, 1888 (Fed. Cir. 1991) (citations omitted). That is, something in the prior art as a whole must suggest the desirability, and thus the obviousness, of making the combination. **See In re Beattie**, 974 F.2d 1309, 1312, 24 USPQ2d 1040, 1042 (Fed. Cir. 1992); **Lindemann Maschinenfabrik GmbH v. American Hoist and Derrick Co.**,

Appeal No. 1999-0080  
Application No. 08/558,929

730 F.2d 1452, 1462, 221 USPQ 481, 488 (Fed. Cir. 1984). In determining obviousness/nonobviousness, an invention must be considered "as a whole," 35 U.S.C. § 103, and claims must be considered in their entirety. **Medtronic, Inc. v. Cardiac Pacemakers, Inc.**, 721 F.2d 1563, 1567, 220 USPQ 97, 101 (Fed. Cir. 1983).

Since the limitation that "said enabler file being initially stored in a read-write memory device so that said enabler file may be replaced with an updated enabler file when system changes are made in said computing system" is not clearly taught or fairly suggested by the combination of Sherer and Arnold, we will not sustain the rejection of claim 21. Since claims 22-26 contain similar limitations which are not taught or suggested by Sherer and Arnold, we will not sustain the rejection of claim 22-26 and the dependent claims 2-6, 11-17 and 27. The examiner relies upon Schmidt and Mitani for various limitations in dependent claims, but does not rely on these teachings for the subject matter lacking as discussed above and these teachings do not remedy the deficiency in the combination.

Appeal No. 1999-0080  
Application No. 08/558,929

**CONCLUSION**

To summarize, the decision of the examiner to reject claims 2-6, 11-17 and 21-27 under 35 U.S.C. § 103 is reversed.

**REVERSED**

JOSEPH L. DIXON	)	
Administrative Patent Judge	)	
	)	
	)	
	)	
	)	BOARD OF PATENT
ANITA PELLMAN GROSS	)	APPEALS
Administrative Patent Judge	)	AND
	)	INTERFERENCES
	)	
	)	
	)	
STUART S. LEVY	)	
Administrative Patent Judge	)	

jld/vsh

Appeal No. 1999-0080  
Application No. 08/558,929

JOHN S. FERRELL  
2225 EAST BAYSHORE BLVD.  
SUITE 200  
PALO ALTO, CA 94303