



(72) LAPIE, Ronan, FR

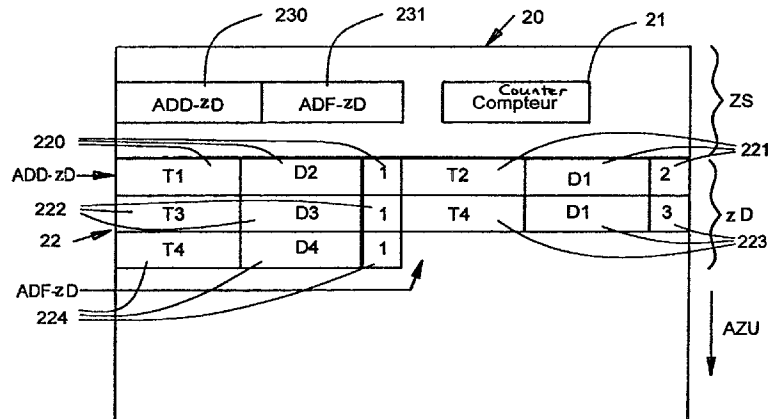
(71) BULL CP8, FR

(51) Int.Cl.⁶ G06F 11/22

(30) 1996/12/24 (96/15997) FR

(54) **TERMINAL ET PROCEDE D'AUTODIAGNOSTIC OU DE SUPERVISION ET OBJET PORTATIF UTILISE DANS UN TEL TERMINAL OU PROCEDE**

(54) **SELF-DIAGNOSTIC OR SUPERVISORY TERMINAL AND METHOD AND PORTABLE OBJECT USED IN SUCH TERMINAL OR METHOD**



(57) Terminal doté d'un programme d'application, d'au moins une sortie constitué soit par un affichage, soit par une imprimante, soit par un réseau de communication, soit par un objet portable et coopérant avec un objet portable doté d'une zone de mémoire non-volatile (ZD) contenant des données et comportant un lecteur communiquant avec ledit objet portable, caractérisé en ce que l'appareil comporte des moyens de lecture ou de stockage, dans sa mémoire, de données (Ti, Dj, Sk) d'autodiagnostic ou de supervision et un moyen d'émission desdites données vers des sorties (1-4) spécifiées en fonction d'informations fournies par les données d'autodiagnostic ou de supervision suite à l'exécution d'au moins une tâche Tt de son programme d'application en relation avec l'objet portable.

(57) The invention concerns a terminal equipped with an application programme, with at least one output consisting of either a display or a printer, or a communication network, or a portable object and cooperating with a portable object equipped with a non-volatile memory zone (ZD) containing data and comprising a reader communicating with said portable object. The invention is characterised in that the apparatus comprises means for reading or storing in its memory self-diagnostic or supervisory data (Ti, Dj, Sk) and means for transmitting said data to specific outputs (1-4) on the basis of information supplied by the self-diagnostic or supervisory data following the execution of at least one task (Tt) of its application programme relative to the portable object.

ABSTRACT

Inventor: Ronan LAPIE

Terminal equipped with an application program, with at least one output constituted either by a display, or by a printer, or by a communications network, or by a portable object, and cooperating with a portable object equipped with a non-volatile memory area (ZD) containing data, and comprising a reader which communicates with said portable object, characterized in that the device comprises means for reading or storing, in its memory, self-diagnostic or supervisory data (Ti, Dj, Sk) and means for sending said data to outputs (1-4) specified as a function of information supplied by the self-diagnostic or supervisory data following the execution of at least one task Tt of its application program in connection with the portable object.

1 TERMINAL AND PROCESS FOR SELF-DIAGNOSIS OR SUPERVISION
2 AND PORTABLE OBJECT USED IN A TERMINAL OR PROCESS OF THIS TYPE
3

4 The present invention relates to a terminal and a process
5 for self-diagnosis or supervision and the portable object of the
6 microcircuit card type used in a process or terminal or reader of
7 this type. These objects comprise a central processor, a program
8 memory containing an executable code constituting the operating
9 system, a non-volatile and programmable data memory, and one or
10 more communication interfaces. The terminals are devices equipped
11 with an interface compatible with those of the objects; they have
12 a central processor and software that is capable of communicating
13 and operating on the data issuing from the non-volatile memory of
14 the portable object.

15 In general, the terminals are equipped with specific
16 software corresponding to their utilization, for example portable
17 payment terminals are equipped with an operating program of the
18 banking type. This software is produced or specified by the
19 institution that manages this application; in the example cited,
20 this is a banking institution. This institution is generally not
21 the manufacturer of the terminal; it purchases or orders the
22 manufacture of the hardware part, that is the terminal, and it
23 installs the specific program into it in order to configure its
24 terminal for its own application. The banking institution thereby
25 has the advantage of purchasing a standard, and therefore
26 inexpensive, product and adapting it according to its needs. The
27 manufacturer offers a basic model which can be suited to a
28 plurality of applications, which enables it to expand its market.

29 The institution running a given application may wish to use
30 a plurality of card reader terminal models; it is not desirable
31 to develop application software for each of the terminals. This
32 has led manufacturers to implement a basic software layer that
33 ensures the interface between the hardware and the application
34 software. This software layer enables the same application
35 software to be adapted to different terminals. One way to do this
36 is to create an interpreter so that the institution can develop

1 its application in a well-known high-level language, nearly
2 independent of the constraints of the hardware. Another way to do
3 this is to set up a low-level software layer that manages all the
4 hardware input-outputs and to make available to the operating
5 institution a library of primitives that the application software
6 will call.

7 In all cases, it must be possible to validate or to test the
8 terminal in its entirety. The validation or testing of the
9 terminal must take into account both parts: the hardware with its
10 basic software and the application software. A self-test makes it
11 possible to verify each device of the terminal; it is generally
12 constituted by a routine implemented in the basic software. The
13 testing of the application software must be done in the
14 laboratory; it is important in this type of application to
15 properly verify the operation of the program before it is placed
16 in service. However, the multiplicity of cards results in a very
17 large number of specific cases that are not reproducible in the
18 laboratory. The object of the following invention is to validate
19 or test, under normal usage conditions, the operation of
20 application software.

21 To this end, the invention relates to a terminal equipped
22 with an application program, with at least one output constituted
23 either by a display, or by a printer, or by a communication
24 network, or by a portable object, and cooperating with a portable
25 object equipped with a non-volatile memory area containing data,
26 and comprising a reader which communicates with this portable
27 object, characterized in that the device comprises means for
28 reading or storing in its memory diagnostic or supervisory data
29 and means for sending these data to outputs specified as a
30 function of information supplied by the self-diagnostic or
31 supervisory data following the execution of at least one task of
32 its application program in connection with the portable object.

33 According to another characteristic, the means for sending
34 the self-diagnostic data are activated a certain number of times.

35 According to another characteristic, the means for sending

1 the self-diagnostic or supervisory data comprise means for
2 writing in the portable object connected to the device.

3 According to another characteristic, the self-diagnostic or
4 supervisory data are constituted by at least one triplet of
5 information corresponding, for a first piece of information, to a
6 predetermined task of the application program, for the second
7 piece of information to a data type correlated to the task
8 executed and to be presented to an output, and for the third
9 piece of information to a value for specifying the output to
10 which the data type must be presented among those present in the
11 terminal.

12 According to another characteristic, the device has a means
13 for testing for the presence of self-diagnostic or supervisory
14 data in a portable object and for initiating the reading and the
15 storage of these data in a specific area ZTD of the memory of the
16 terminal.

17 According to another characteristic, the terminal comprises
18 means for entering self-diagnostic or supervisory data into a
19 portable object.

20 Another object of the invention is to propose a process for
21 supervising the operation of a terminal or for the self-diagnosis
22 of a terminal.

23 This object is achieved due to the fact that the self-
24 diagnostic or supervisory process, from at least one triplet of
25 information corresponding, for a first piece of information, to a
26 predetermined task of an application program executed either by a
27 portable object or by a terminal, for the second piece of
28 information to a data type correlated to the task executed and to
29 be presented to an output, and for the third piece of information
30 to a value for specifying the output among those present in the
31 terminal, characterized in that it is comprised of:

32 - executing a task of the application program in the
33 terminal;

34 - testing an indicator either in the terminal or in the
35 portable object to determine whether a self-diagnostic or

1 supervisory function is operational, then in the case of a
2 positive response;

3 - searching in the memory of either the portable object or
4 the terminal to see if among the triplets of information stored
5 there is a triplet wherein the first piece of information
6 corresponds to the predetermined task executed by the terminal or
7 the card;

8 - sending to the output specified by the triplet thus read
9 the value of the datum correlated to the task executed and to be
10 labelled by the second piece of information in the triplet.

11 According to another characteristic, the process comprises a
12 testing step comprised of determining whether there are other
13 tasks to be executed, and following the execution of these tasks,
14 searching for all of the triplets of information corresponding to
15 the execution of this task.

16 According to another characteristic, the process comprises a
17 step for reading from a portable object storing in its non-
18 volatile memory a plurality of triplets and a step for storing
19 these triplets in a non-volatile memory area of the terminal,
20 followed by a step for activating an indicator of an active self-
21 diagnostic or supervisory function.

22 According to another characteristic, the process comprises a
23 testing step for determining whether the portable object is a
24 card specific to the self-diagnostic or supervisory function or a
25 so-called general-purpose card.

26 According to another characteristic the self-diagnostic or
27 supervisory data are constituted by a fourth field of information
28 containing in the portable object initially the write address
29 (Adr-V), the number of octets to be written (Nb-V), and after the
30 self-diagnosis operation, the value to be written (Val).

31 Another object of the invention is to propose a portable
32 object that can be used with the terminal and the self-diagnostic
33 or supervisory process.

34 This object is achieved due to the fact that the portable
35 object is a microprocessor card operating by means of an

1 operating system stored in the card and comprising a non-volatile
2 memory containing at least one triplet of information in a
3 predetermined area of this non-volatile memory whose location is
4 defined by address fields located in the memory part used to
5 store the operating system.

6 According to another characteristic, the part of non-
7 volatile memory used to store the operating system also
8 comprises, in a memory field, a piece of information constituting
9 a counter of utilizations of the self-diagnostic function.

10 According to another characteristic, the memory area storing
11 the operating system comprises a field which makes it possible to
12 store an indicator of the activation of the self-diagnostic or
13 supervisory function.

14 Other characteristic and advantages of the present invention
15 will become more clearly apparent with the reading of the
16 following description given in reference to the appended
17 drawings, in which:

18 Fig. 1 is an explanatory table of the triplets constituted
19 by elementary tasks, data types and output types that can be
20 associated with the execution of each of the tasks of an
21 application program;

22 Fig. 2 is a diagram of the non-volatile memory areas used in
23 a portable testing object necessary to the implementation of the
24 process of the invention;

25 Fig. 3 shows the various steps in the execution of the
26 programs for initializing the terminal and executing the self-
27 diagnostic function;

28 Fig. 4 shows schematic view of the areas for storing
29 information in the non-volatile memory of a so-called general-
30 purpose portable object;

31 Fig. 5 shows the various steps in the execution of a program
32 for initializing the terminal and executing the self-diagnostic
33 function in this terminal with a so-called general-purpose card.

34 One way to implement the invention is comprised, in a first
35 variant, of using a microprocessor card operating by means of an

1 operating system stored in the card and constituting a "so-called
2 smart card" that is pre-initialized with the self-diagnostic
3 data, and of having these data taken into account by the terminal
4 when intending to test the application software.

5 An application software in a terminal can be broken down
6 into elementary tasks which occur at predetermined moments. For
7 example, for a banking application, a transaction can be broken
8 down into the following elementary tasks (Ti): Verification (T1,
9 Fig. 1) of whether the card inserted is authorized,
10 Authentication (T2) of the bearer, Acquisition (T3) of the data
11 of the transaction in the terminal, Writing of these data into
12 the microcircuit of the card, Validation (T4) of the transaction
13 in the terminal and the card.

14 In addition, and always during a transaction, the
15 application software manipulates data; these data can be used
16 temporarily, like the Code created by the bearer (Cp), which is
17 stored in the memory of the terminal, or the identity of the
18 bearer (Ip), which is stored in the card, or like the Amount of
19 the transaction (Mt) or the Date of the transaction (Dt), which
20 are stored in the terminal and the card. During the execution of
21 each elementary task, each of these data may be initialized,
22 modified or unchanged. The self-diagnostic utility function of
23 the application software is comprised of verifying the data of
24 the transaction at the time of certain tasks, under normal usage
25 conditions. This function can be performed either by the basic
26 software or by the application software.

27 To do this, the operator responsible for the test develops a
28 grid composed, on one side, of the identifiable elementary tasks,
29 designated Ti, and on the other side of data Dj constituted by
30 information such as: Cp, Ip, Mt and Dt. Fig. 1 shows an example
31 of such a grid. In order to test the correct execution of a
32 transaction, the operator chooses to verify the value of certain
33 data during the execution of specific tasks. This involves
34 associating a datum Dj with a task Ti; these associations are
35 symbolized by crosses in the grid of Fig. 1. A third piece of

1 information S_k is added. The value of this code indicates the
2 type of output used to which the datum to be verified is to be
3 sent: to the network when S_k is at a first value, for example "1"
4 ($S_k = 1$), to the printer when S_k is at a second value, for
5 example "2" ($S_k = 2$), or to the screen when S_k is at a third
6 value, for example "3" ($S_k = 3$). The operator enters the triplets
7 of information (T_i, D_j, S_k) into a special so-called diagnostic
8 central processor, which central processor is equipped with a
9 card reader. The diagnostic software is configured in accordance
10 with the application to be tested so that the triplets ($T_i, D_j,$
11 S_k), when captured, are identified on the screen by the precise
12 indication of the elementary tasks and the data to be verified,
13 and not the numeric labels T_i, D_j, S_k .

14 The card containing the self-diagnostic data is either a
15 special card or a general-purpose card normally used for an
16 application. A detailed description of an embodiment is given for
17 each case.

18 The first case described is that in which a special card
19 called a "test card" (20, Fig. 2) is used to contain the self-
20 diagnostic data. A security procedure is implemented to prevent a
21 defrauder from being able to use a card of this type in an
22 unauthorized manner. The test card contains in the secret memory
23 area, not shown, a secret so-called diagnostic code "KD." This
24 secret code must first be presented to the card, which verifies
25 it and, if it is equal to a reference code, authorizes the
26 writing of the self-diagnostic data into the programmable memory
27 of the card.

28 While providing for the storage of the self-diagnostic data,
29 the non-volatile programmable memory of the test card also has,
30 in addition to the system area ZS which contains the operating
31 system of the card and the other usable area (AZU) which allows
32 other types of storage, an area (22) called "ZD." It is in this
33 area that the triplets (T_i, D_j, S_k) are stored in succession.
34 Thus, a first area (220) of a memory allows the storage of the
35 first triplet $T_1, D_2, 1$; a second area (221) allows the storage

1 of the second triplet T2, D1, 2; a third area (222) allows the
2 storage of the third triplet T3, D3, 1; a fourth area (223)
3 allows the storage of the fourth triplet D4, D3, 3; a fifth area
4 (224) allows the storage of the triplet T4, D4, 1; T1, T2, T3,
5 T4, D1, D2, D3, D4 respectively representing the information in
6 Fig. 1. It is obvious that the portable object can comprise more
7 or fewer triplets depending on the type of supervision or self-
8 diagnosis that it is desirable to perform on the tasks executed
9 by the application program. The area ZD is labelled by its start
10 address "ADD_ZD" and its end address "ADF-ZD"; the two address
11 values are stored in the part (230, 231) of the programmable
12 memory allocated to the operating system.

13 The non-volatile programmable memory is of the EPROM,
14 EEPROM, FeRAM, SRAM or FLASH type. Fig. 2 describes the
15 organization of this memory using the information cited in Fig.
16 1. Advantageously, the datum Dj is the physical address of the
17 datum to be verified in the working memory of the terminal.

18 Once programmed, the test card is inserted into the terminal
19 in which the self-diagnostic function must be run. Fig. 3 is a
20 flow chart illustrating the chronology of the events of the
21 program, constituted by a wait and test sequence (1, 2, 3), the
22 test triggering, as a function of the result, either a sequence
23 for loading the terminal with the self-diagnostic data (4 through
24 7, Fig. 3), or a sequence for executing the self-diagnostic
25 program (8 through 16, Fig. 3), which is incorporated either into
26 the basic software of the terminal or into the application
27 software. Step 1 is the initialization of the terminal after it
28 is powered up, and step 2 is the phase for waiting for an order
29 or an insertion of a card. In step 3, the terminal tests whether
30 the card inserted into the reader is a general-purpose card, and
31 in step 4 whether the card is a test card. In the latter case,
32 the terminal executes in step 5 a procedure for authenticating
33 the card by means of a reference code or by means of a standard
34 challenge-response authentication schema using an algorithm and a
35 secret key (KD).

1 Once the test card has been identified and authenticated,
2 the program of the terminal reads in step 6 the information
3 contained in the area ZD. The selection and the location of the
4 triplets are performed with the aid of the two pointers ADD_ZD
5 and ADF-ZD. The triplets (Ti, Dj, Sk) read successively in the
6 area ZD are stored in the same order in an area of the memory of
7 the terminal called ZTD. Once the last triplet has been stored in
8 the area ZTD, the terminal program, in step 7, sets a self-
9 diagnostic indicator "Ind_DT" in the memory of the terminal to
10 the active state. Then the terminal program loops back to wait
11 for a command or another insertion of a card in step 2.

12 A new card is inserted; it is a general-purpose card
13 compatible with the application run by the terminal. As stated
14 above, the application software in the terminal is broken down
15 into elementary tasks Tt which can be executed individually (step
16 8). At the end of the execution of each task, which can be
17 labelled by the code Tt, the application program, in step 9,
18 tests the indicator Ind_DT of the terminal. If it is inactive,
19 the self-diagnostic function is not operational, and the program
20 continues to execute the other tasks. If the indicator Ind_DT is
21 active, the program of the terminal, in step 10, searches the
22 area ZTD of the memory of the terminal for the first triplet (Ti,
23 Dj, Sk) for which Tt = Ti, that is, to see if there is a datum to
24 be tested as a result of the task that has just been executed. If
25 yes, in step 11, the value "Val" of this datum (Dj) is
26 temporarily stored in the memory of the terminal, and as a
27 function of the value of Sk, it is processed in the following
28 way:

29 If Sk is equal to "1" (step 12), the datum Dj must be sent
30 to the network. In this case, a block of three data is
31 constituted by the value of the field Tt, the label Dj of the
32 datum to be analyzed and the value "Val" of this datum extracted
33 from the memory of the terminal. These blocks are stored one
34 after another in an area of the memory of the terminal called
35 "ZDR." The content of this area is sent to the network at the end

1 of the transaction or when there is a request by the network for
2 self-diagnostic data. Once all the data have been transmitted,
3 the area ZDR is cleared so that it can be reused when a there is
4 new insertion of a card.

5 If Sk is equal to "2" (step 13), the datum Dj must be sent
6 to the printer of the terminal for printing. In this case, a
7 message is created in the software buffer of the printer; it is
8 composed of a text (ASCII code) indicating the nature of the
9 datum, for example "AMOUNT," followed by the decimal or
10 hexadecimal value of the datum Dj; the message ends with a
11 separator and a "CARRIAGE RETURN - SKIP LINE." It is possible to
12 gather all of the self-diagnostic messages and print them at the
13 end of the transaction.

14 If Sk is equal to "3" (step 14), the datum must be sent to
15 the display unit of the terminal for display. In this case, a
16 message is created in the buffer of the display unit, composed of
17 a text (ASCII code) indicating the nature of the datum, for
18 example "AMOUNT," followed by the decimal or hexadecimal value of
19 the datum Dj. The messages corresponding to each element (Tt, Dj,
20 "3") are successively displayed for a certain time set by the
21 program. It is possible to gather all of the messages and display
22 them at the end of the transaction; the scrolling of the messages
23 can be controlled by pressing a key on the keyboard of the
24 terminal.

25 Once the datum Dj has been processed, the program verifies
26 in step 15, whether there are other triplets in ZTD for which T_i
27 = T_t . If yes, the program loops back to step 11 and processes a
28 new triplet. For each elementary task, the search for triplets is
29 performed by scanning the entire area ZTD. If there are no more
30 triplets ($T_i=T_t$, Dj, Sk) to be processed, the program, in step
31 16, continues in sequence and may, in a variant, proceed to
32 another task without executing the steps 17 and 18 described
33 below. If there are no other tasks to be executed, the program
34 loops back to step 3, to wait for a command or for a new
35 insertion of a card.

1 It is possible to associate a counter initialized with a
2 certain number with the indicator Ind-DT in the terminal so that
3 the self-diagnostic function is only activated for this number of
4 general-purpose card insertions. In order to do this, the
5 operator has pre-entered this number into a specific location
6 (21, Fig. 2) of the programmable memory of the test card, for
7 example, next to the locations (230, 231) of ADD_ZD and ADF_ZD.
8 In this case, this number is stored in the memory of the terminal
9 after the insertion of the test card in step 6. Then this number
10 is decremented (step 17) at the end of each execution of a self-
11 diagnostic function (a YES output from step 16). When it reaches
12 "0," the indicator Ind_DT is set to the inactive position (step
13 18) and the content of the area ZTD is possibly erased.

14 If the counter is not installed, steps 17 and 18 do not
15 exist and the self-diagnostic function is executed only once or
16 indefinitely until a new insertion of the test card switches the
17 indicator Ind_DT to the inactive position.

18 It is possible to avoid the utilization of a test card and
19 to use only general-purpose cards, on condition that they support
20 the special self-diagnostic functions. For this reason, the
21 programmable memory of the general-purpose card contains, in
22 addition to the system area ZS and the user area ZU, an area ZD
23 which is labelled by its start address "ADD_ZD" and its end
24 address "ADF-ZD" (see Fig. 4). The programmable memory of the
25 general-purpose card also contains in its system area, in a
26 location (232), an indicator "Ind_D" which indicates whether or
27 not the self-diagnostic function is active. All of these data
28 ADD-ZD, ADF-ZD, Ind-D are stored in locations (230, 231, 232) of
29 the part ZS of the programmable memory allocated to the operating
30 system. The two address values are determined and written into
31 the area ZD during the customization of the card; this method is
32 simple to implement but has the drawback of requiring the
33 reservation of a sizeable location in all cards that can be used
34 for the self-diagnosis.

35 Advantageously, the location in the area ZD can be allocated

1 dynamically by the operating system of the card after the correct
2 entry of the code KD. The operator indicates to the card the
3 number of triplets (T_i , D_j , S_k) or the number of octets to be
4 reserved for ZD. The operating system of the card then searches
5 in the programmable memory for a blank location of sufficient
6 size. If the memory does not contain any such blank location, the
7 operating system returns an error message and interrupts the
8 procedure for entering the self-diagnostic data. In the opposite
9 case where there is sufficient space, the operating system stores
10 the start address "ADD_ZD" and the end address "ADF-ZD." It will
11 be seen below how, after the execution of the self-diagnostic
12 function, it is possible to erase the presence of the area ZD,
13 thus releasing this memory space.

14 The same is true for the test card. A security procedure is
15 provided in order to prevent a defrauder from being able to use a
16 general-purpose card to enter self-diagnostic data. A mechanism
17 of the challenge-response type with an algorithm and a secret
18 code makes it possible to authenticate the operator and authorize
19 the writing and reading (it will be seen why below) of the
20 triplets in ZD.

21 In the case where a general-purpose card is used to transmit
22 the self-diagnostic data, the code S_k can assume a fourth value
23 4; this value indicates that the value of the datum D_j to be
24 verified has been written into the card. In this case, a fourth
25 field located at an address "Adr-V" is allocated at the end of
26 the triplet (T_i , D_j , $S_k=4$), and thus quadruplets are stored. The
27 size of this field corresponds to that of the data to be written;
28 the operator must therefore specify the number of octets "Nb-V"
29 in this fourth field, and its content is initially constituted by
30 the write address (Adr-V), then after the output by the value
31 "Val," as seen below. The fifth triplet (225) of Fig. 4 has this
32 structure. When all of the triplets (T_i , D_j , S_k) (220 through
33 225) have been entered into the area ZD, the indicator "Ind_D" is
34 set to the active position, thus indicating that the self-
35 diagnostic function is active in this card.

1 Fig. 5 shows the sequence of operations when the card
2 described above is inserted into a terminal. Step 1 is the
3 initialization of the terminal after it is powered up and step 2
4 is the phase for waiting for the insertion of a card; the program
5 continues when the card is recognized as being compatible with
6 the application through the terminal's recognition of the
7 presence of the necessary information. In this step 2, the
8 program performs the selection of the entity corresponding to the
9 application. Unlike the test card, when the general-purpose card
10 is inserted by the bearer, the latter can be completely unaware
11 that the self-diagnostic function is active.

12 In step 3, the terminal tests whether the indicator Ind_D in
13 the card is set to the active position and thus whether the self-
14 diagnostic function is operational. The indicator can be sent
15 either by a particular value in the octets transmitted by the
16 card during the power-up phase, or by a particular value
17 transmitted during the selection of the entity corresponding to
18 the application used in the card. If Ind_D is active, the program
19 proceeds to step 4. During this step 4, the area ZD is read with
20 the aid of the two address values ADD_ZD and ADF_ZD and all of
21 the triplets read in the card are stored in the memory ZTD of the
22 terminal. If the triplet comprises a datum Sk whose value is "4,"
23 the operating system of the card returns, in addition to the
24 three values Ti, Dj and Sk, the address "Adr-v" of the fourth
25 field reserved for writing the datum into ZD and the number of
26 octets "Nb_v" in this field. For security reasons, read access to
27 the area ZD of the card is only granted by the operating system
28 of the card if the indicator Ind_D of the card is active. Once
29 all of the information contained in the area ZD has been stored
30 in the memory area ZTD of the terminal, the terminal sets its
31 self-diagnostic indicator Ind-D to the active position (step 5).
32 Steps 3, 4 and 5 are parts of the sequence for initializing the
33 dialogue between the general-purpose card and the terminal and
34 are executed before the execution of the application program.

35 As stated above, the application software is broken down

1 into elementary tasks which can be tested individually. At the
2 end of the execution of each task, which can be labelled by a
3 number Tt (step 6) for example the basic software resumes control
4 and tests whether the diagnostic indicator inside the terminal is
5 active (step 7). If it is active, in step 8 the program searches
6 to see if there is an element (Ti, Dj, Sk) stored in the area ZTD
7 that has a field value Ti equal to that of Tt. If yes, there is a
8 datum (labelled Dj) to be verified as a result of the task that
9 has just been executed; the value of this datum is then
10 temporarily stored in the memory of the terminal. As a function
11 of the value of Sk, it is processed in the following way (step
12 9):

13 If Sk is equal to "1" (step 10), the datum Dj must be sent
14 by the terminal to the network. A block of three data is then
15 constituted by the value of the field Tt, the label Dj of the
16 datum to be analyzed and the value "Val" of this datum extracted
17 from the memory of the terminal. These blocks are stored one
18 after another in an area of the memory of the terminal called
19 "ZDR." The content of this area is sent to the network at the end
20 of the transaction or when there is a request from the network
21 for the self-diagnostic data. Once all of the data has been
22 transmitted, the area ZDR is cleared so that it can be reused
23 when a there is a new insertion of a card.

24 If Sk is equal to "2," the datum must be sent to the printer
25 of the terminal, and the program continues with step 11. During
26 this step 11 a message is created in the buffer of the printer,
27 composed of a text (ASCII code) indicating the nature of the
28 datum, for example "AMOUNT," followed by the decimal or
29 hexadecimal value of the datum Dj, and the message ends with a
30 separator and a "CARRIAGE RETURN - SKIP LINE." Advantageously, it
31 is possible to gather all of the messages and print them at the
32 end of the transaction.

33 If Sk is equal to "3," the datum must be sent to the display
34 unit of the terminal, and the program continues with step 12.
35 During this step 12, a message is created in the buffer of the

1 display unit, composed of a text (ASCII code) indicating the
2 nature of the datum, for example "AMOUNT," followed by the
3 decimal or hexadecimal value of the datum Dj. The messages
4 corresponding to each element (Tt, Dj, "3") are successively
5 displayed for a certain time set by the self-diagnostic software.
6 Advantageously, it is possible to gather all of the messages and
7 display them at the end of the transaction; the scrolling of the
8 messages can be controlled by pressing a key on the keyboard of
9 the terminal.

10 If Sk is equal to "4," the datum must be stored in the card,
11 and the program continues with step 13. During this step 13, a
12 fourth field is reserved for this purpose in ZD, the address
13 "Adr-v" of this fourth field and the number of octets "Nb_v" in
14 this field having been stored in the area ZTD during the loading
15 of the self-diagnostic data into the terminal. The terminal then
16 sends the card a write command with the following parameters:

17 Write address: Adr-v

18 Number of octets to be written: Nb-v

19 Value to be written: "Val" of the datum Dj.

20 The write operation in the area ZD is only authorized by the
21 operating system of the card when the fourth field of a triplet
22 of the Sk=4 type corresponds to the Ti. In the case where the
23 triplet Ti of the card is not of the Sk=4 type, the write
24 operation is denied. An execution report is systematically
25 returned to the terminal after each write command; if the latter
26 has not been successful, the terminal warns the user with a
27 message. The utilization of the stored data is explained below. A
28 variant is comprised of temporarily storing all the values of the
29 data Dj of the Sk=4 type and of executing the commands for
30 writing the values at the end of the transaction.

31 Once the datum Dj has been processed, in step 14, the
32 program verifies whether there are other triplets in the area ZTD
33 for which Tt=Ti. If yes, the program loops back to step 9 and
34 processes a new triplet. For each elementary task, the search for
35 triplets is performed by scanning the entire area ZTD. If there

1 are no more triplets ($T_i=T_t$, D_j , S_k) to be processed, the program
2 proceeds to step 15, searching for another task to be executed.
3 If there are no other tasks to be executed, the program loops
4 back to step 2 to wait for a new insertion of a card.

5 In the case where a general-purpose card is used to transmit
6 the self-diagnostic data, the self-diagnostic function should be
7 able to be executed only once. In effect, the operator may intend
8 to perform only one test of the card reader terminal, after which
9 the data must not leave the terminal. Moreover, if the datum to
10 be verified has a field S_k of the 4 type, a single storage is not
11 possible. In order to start the function another time, the card
12 must be reprogrammed by the operator. To prevent it from being
13 used several times with the self-diagnostic function, just after
14 the reading of the area ZD by the terminal, the operating system
15 of the card will set the indicator Ind_D to the inactive state.
16 For security reasons, it can also erase all the triplets of the
17 $S_k=1, 2$ and 3 types. The indicator $Ind-D$ is inactive, and the
18 reading of the area ZD is no longer possible.

19 The data corresponding to the type $SK=4$ are processed when
20 the general-purpose card enters a terminal authorized to read it,
21 that is, a terminal that is authenticated in the same way as when
22 the self-diagnostic data are written. Once all the triplets have
23 been read, the total erasure of the area ZD can be carried out.
24 This erasure can be triggered by a special command or during the
25 first write operation in the area ZD. The erasure, which is
26 justified for security reasons, makes it possible to release the
27 space occupied by the area ZD. This location can be used by the
28 application.

29 This "one-shot" style of operation of the self-diagnostic
30 function can be advantageous when a person reports his credit
31 card to a bank branch, declaring that his card "does not work" in
32 a certain type of payment terminal. The branch will store the
33 self-diagnostic data in this card, specifying that the data of
34 the transaction (amount, date, certificate value) are to be
35 written into the card by setting the third field S_k , in each

1 triplet corresponding to a task to be recorded, at the value "4"
2 (Sk=4). The person returns to the merchant where the terminal to
3 be tested is located, executes a transaction and returns to his
4 branch, which analyzes the information stored in ZD, or has it
5 analyzed remotely.

6 One-shot operation is also advantageous for verifying the
7 operation of a terminal suspected of fraud. A banking institution
8 discovers that transactions in a terminal have been credited
9 without there having been any request to debit client accounts.
10 The banking institution sends inspectors provided with general-
11 purpose cards with the self-diagnostic function. Upon their
12 return, the data loaded into the card are analyzed.

13 Another example: a banking institution may find it
14 advantageous to quickly learn the time and the place where the
15 card is used for the first time. For this reason, before the card
16 is delivered to its bearer, it contains two triplets of the Sk=1
17 type, in which each third field Sk is at the value "1." In the
18 area ZD, with the datum Dj corresponding to the date of the
19 transaction and the datum Dj corresponding to the identity of the
20 terminal. During the first transaction, the two blocks (Ti, Dj,
21 "date") and (Ti, Dj', "terminal ID") will be sent to the network
22 immediately.

CLAIMS

1
2
3 1. A terminal equipped with an application program, with
4 at least one output constituted either by a display, or by a
5 printer, or by a communications network, or by a portable object,
6 and comprising a portable object reader, and cooperating through
7 this reader with a portable object equipped with a non-volatile
8 memory area (ZD) containing data, characterized in that it
9 comprises means for reading or storing, in its memory, self-
10 diagnostic or supervisory data (Ti, Dj, Sk) and means for sending
11 said data to outputs (1-4) specified as a function of information
12 supplied by the self-diagnostic or supervisory data following the
13 execution of at least one task Tt of its application program in
14 connection with the portable object.

1 2. The terminal according to claim 1, characterized in
2 that the means for sending the self-diagnostic data are activated
3 a certain number of times.

1 3. The terminal according to claim 1, characterized in
2 that the means for sending the self-diagnostic or supervisory
3 data comprise means for writing into the portable object
4 connected to the terminal.

1 4. The terminal according to claim 1, characterized in
2 that the self-diagnostic or supervisory data are constituted by
3 at least one triplet (Ti, Dj, Sk) of information corresponding,
4 for a first piece of information (Ti) to a predetermined task of
5 the application program, for the second piece of information (Dj)
6 to a data type correlated to the task executed and to be
7 presented to an output, and for the third piece of information to
8 a value (Sk) for specifying the output to which the data type
9 must be presented among those present in the terminal.

1 5. The terminal according to claim 1, characterized in

2 that it has a means for testing for the presence of self-
3 diagnostic or supervisory data in a portable object and for
4 initiation of the reading and the storage of these data in a
5 specific area ZTD of the memory of the terminal.

1 6. The terminal according to claim 1, characterized in
2 that it comprises means for entering self-diagnostic or
3 supervisory data into a portable object.

1 7. A process for supervising the operation of a terminal
2 or for the self-diagnosis of a terminal from a triplet of
3 information corresponding, for a first piece of information, to a
4 predetermined task of an application program executed either by a
5 portable object or by a terminal, for the second piece of
6 information to a data type correlated to the task executed and to
7 be presented to an output, and for the third piece of information
8 to a value for specifying the output among those present in the
9 terminal, characterized in that it is comprised of:

10 - executing (9) a task of the application program in the
11 terminal;

12 - testing (9) an indicator either in the terminal or in the
13 portable object to determine whether a self-diagnostic or
14 supervisory function is operational, then in the case of a
15 positive response;

16 - searching (9) in the memory of either the portable object
17 or the terminal to see if among the triplets of information
18 stored there is a triplet wherein the first piece of information
19 corresponds to the predetermined task executed by the terminal or
20 the card;

21 - sending to the output specified by the triplet thus read
22 the value of the datum correlated to the task executed and to be
23 labelled by the second piece of information of the triplet.

1 8. The process according to claim 7, characterized in that
2 it comprises a testing step (16) comprised of determining whether

3 there are other tasks to be executed, and following the execution
4 of these tasks, of searching for all the triplets of information
5 corresponding to the execution of this task.

1 9. The process according to claim 7, characterized in that
2 it comprises a step for reading from a portable object storing in
3 its non-volatile memory a plurality of triplets and a step (6)
4 for storing these triplets in a non-volatile memory area of the
5 terminal, followed by a step (7) for activating an indicator of
6 an active self-diagnostic or supervisory function.

1 10. The process according to claim 7, characterized in that
2 it comprises a testing step (3, 4) for determining whether the
3 portable object is a card specific to the self-diagnostic or
4 supervisory function or a so-called general-purpose card.

1 11. The process according to claim 7, characterized in that
2 the self-diagnostic or supervisory data are constituted by a
3 fourth field of information containing in the portable object
4 initially the write address (Adr-V), the number of octets to be
5 written (Nb-V), and after the self-diagnosis operation, the value
6 to be written (Val).

1 12. A portable object that can be used with a terminal
2 equipped with an application program, with at least one output
3 constituted either by a display, or by a printer, or by a
4 communications network, or by a portable object, means for
5 reading or storing, in its memory, self-diagnostic or supervisory
6 data (Ti, Dj, Sk) and means for sending said data to outputs (1-
7 4) specified as a function of information supplied by the self-
8 diagnostic or supervisory data following the execution of at
9 least one task Tt of its application program in connection with
10 the portable object, characterized in that it comprises a
11 microprocessor card operating by means of an operating system
12 stored in the card and comprising a non-volatile memory

13 containing at least one triplet (Ti, Dj, Sk) of information in a
14 predetermined area ZD of this non-volatile memory, whose location
15 is defined by address fields located in the memory part used to
16 store the operating system.

1 13. The portable object according to claim 12,
2 characterized in that the part (25) of non-volatile memory used
3 to store the operating system also comprises, in a memory field,
4 a piece of information constituting a counter (21) of
5 utilizations of the self-diagnostic function.

1 14. The portable object according to claim 12,
2 characterized in that the memory area (25) storing the operating
3 system comprises a field which makes it possible to store an
4 indicator (232) of the activation of the self-diagnostic or
5 supervisory function.

		Value of the bearer's code	Identity of the bearer	Value of the transaction	Date of the transaction
		D1 Valeur du code porteur	D2 Identité du porteur	D3 Valeur de la transaction	D4 Date de la transaction
Verification of card authorization T1	Vérification habilitation carte	/ SK	X / SK=1	/	/
Authentication of the bearer T2	Authentification du porteur	X / SK=2	/	/	/
Acquisition of the transaction data T3	Acquisition des données transaction	/	/	X / SK=1	/
Validation of the transaction T4	Validation de la transaction	/	/	X / SK=3	X / SK=1

Fig.1

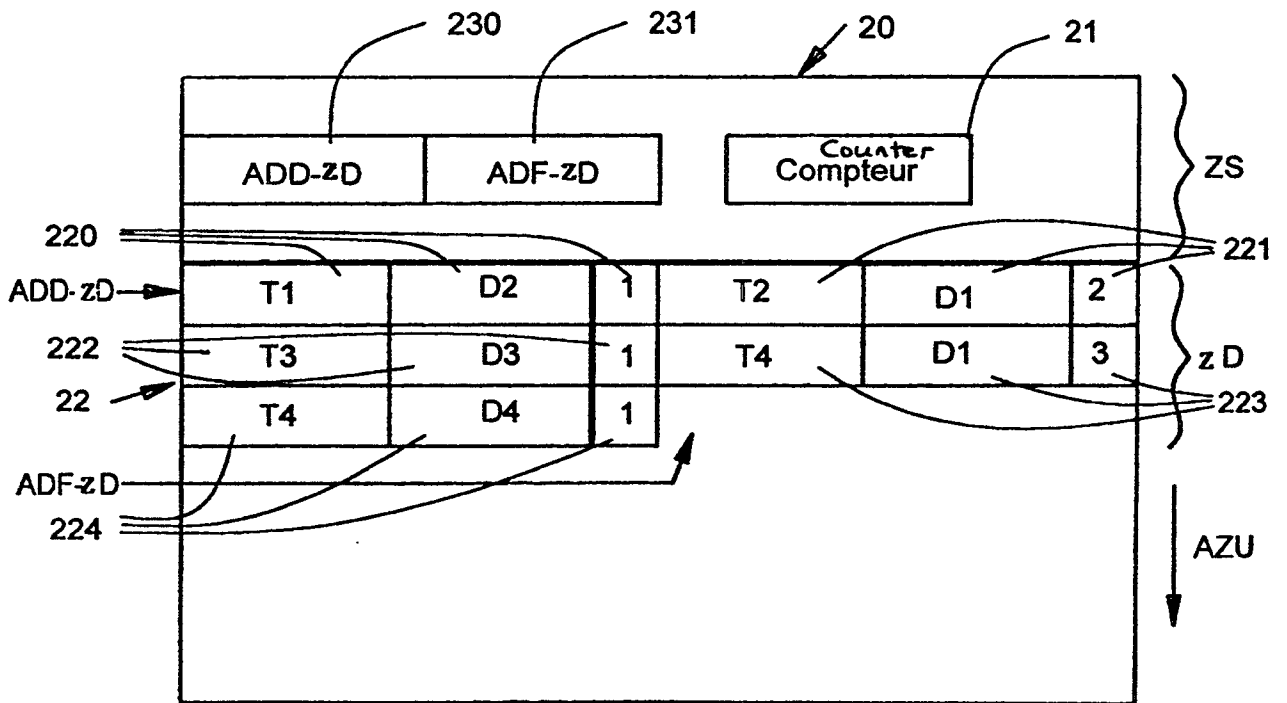


Fig.2

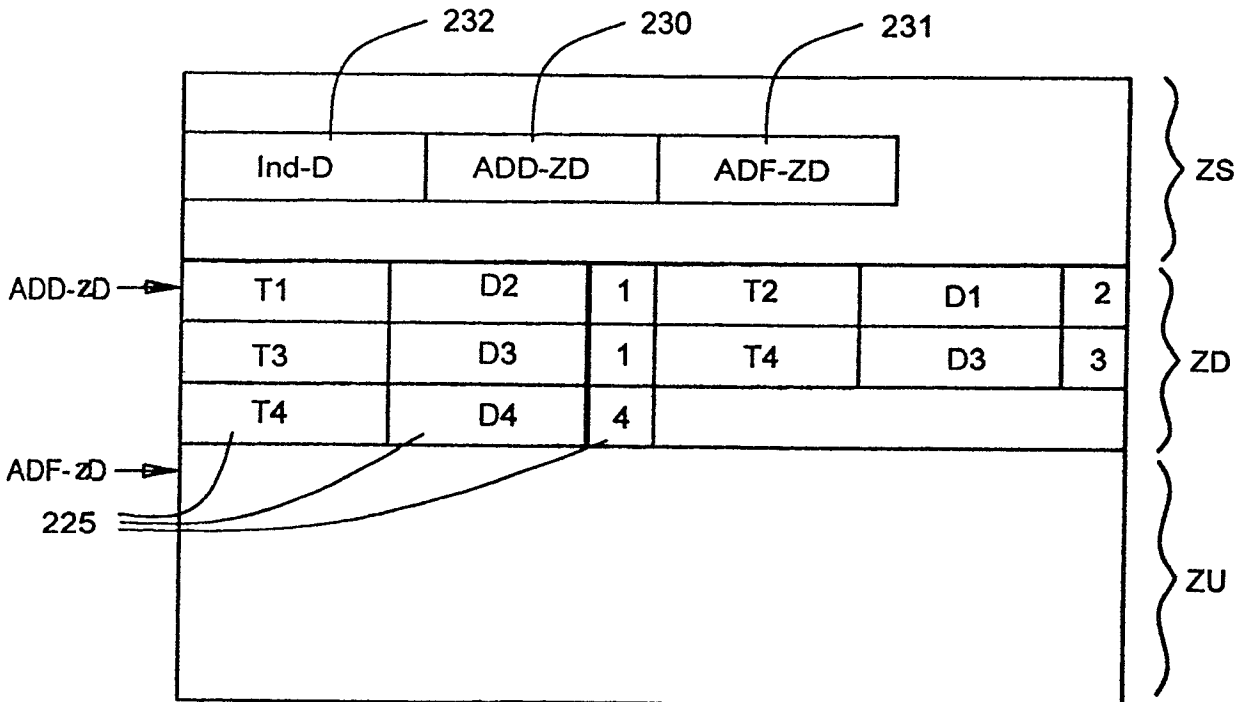


Fig.4

4/4

Powering up of the Terminal

Allumage du Terminal

Fig.5

