

Office de la Propriété Intellectuelle du Canada

Un organisme d'Industrie Canada Canadian Intellectual Property Office

An agency of

Industry Canada

CA 2492367 C 2012/03/27

(11)(21) 2 492 367

(12) BREVET CANADIEN CANADIAN PATENT

(13) **C** 

(22) Date de dépôt/Filing Date: 2005/01/12

(41) Mise à la disp. pub./Open to Public Insp.: 2005/08/09

(45) Date de délivrance/Issue Date: 2012/03/27

(30) Priorité/Priority: 2004/02/09 (EP04002805.2)

(51) Cl.Int./Int.Cl. *G06F 9/44* (2006.01), *G06F 11/32* (2006.01), *G06F 11/34* (2006.01), *G06F 11/36* (2006.01)

(72) Inventeurs/Inventors:

ALBERT, BEATE, DE; KIND, JUERGEN, DE; PAK, IGOR, DE;

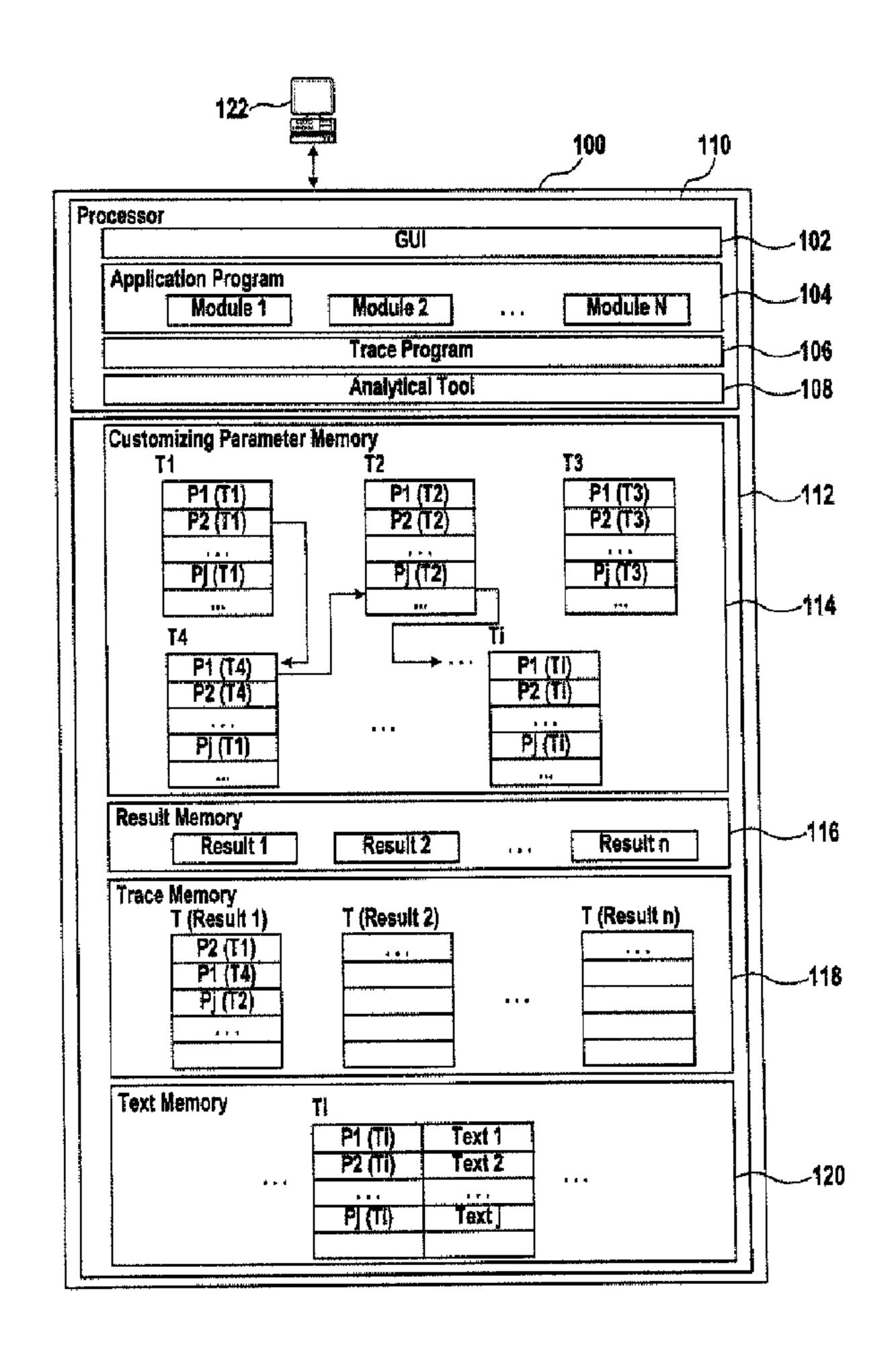
SCHERBERGER, GUENTER, DE

(73) Propriétaire/Owner: SAP AKTIENGESELLSCHAFT, DE

(74) Agent: GOWLING LAFLEUR HENDERSON LLP

(54) Titre: SYSTEME DE TRAITEMENT DE DONNEES

(54) Title: DATA PROCESSING SYSTEM



#### (57) Abrégé/Abstract:

The present invention relates to a method of displaying customising parameters of an application program, the customising parameters being stored in database tables, the method comprising: - execution of the application program, - acquisition of trace





CA 2492367 C 2012/03/27

(11)(21) 2 492 367

(13) **C** 

#### (57) Abrégé(suite)/Abstract(continued):

data during execution of the application program, the trace data comprising a path for each of the results provided by the application program, each path being descriptive of a sequence of database table entries of the database tables holding a sub-set of the customising parameters used for a calculation of the corresponding result, - displaying of the results, - interactively selecting one of the results for further analysis, - displaying the database table entries identified by the path of the selected one of the results.

### <u>Abstract</u>

	•	•
l loto	nraceeina	evetam
Dala	processing	System
-		<b>●</b>

5

The present invention relates to a method of displaying customising parameters of an application program, the customising parameters being stored in database tables, the method comprising:

- execution of the application program,

10

- acquisition of trace data during execution of the application program, the trace data comprising a path for each of the results provided by the application program, each path being descriptive of a sequence of database table entries of the database tables holding a sub-set of the customising parameters used for a calculation of the corresponding result,

15

- displaying of the results,
- interactively selecting one of the results for further analysis,
- displaying the database table entries identified by the path of the selected one of the results.

## 20 (Figure 1)

### Data Processing System

#### Field of the Invention

The present invention relates to the field of data processing systems, and more particularly to acquisition and analysis of trace data.

5

10

15

20

25

30

## **Background and Prior Art**

During the development of a computer program, it often is desirable to test the computer program by providing test data, and then executing the computer program to generate test results. A user then may evaluate the test results and modify the computer program based upon the results. During this process, the user may desire to run at least a portion of the computer program several times to assist, for example, in isolating the suspected cause of an error. When, however, a computer program is very large or has a very long execution time, it may be impractical to replay the entire program for evaluation and debugging. Therefore, a technique called incremental replay was developed which allows a user to select and replay only a portion of the computer program's execution.

To provide effective support for debugging and testing computer programs, a replay apparatus must incur low setup and replay times, it must interfere with and slow down the program's execution as little as possible, and it must not require large amounts of data to be stored. Setup time is the time required to prepare for replay after an original program has been executed at least once. Replay time is the time required to actually re-execute the instructions associated with the desired portion of the computer program.

Some approaches minimize setup time while other approaches minimize replay time. During the setup time, variables or memory locations typically are set to values that are accurate for the portion of the computer program that is to be replayed. Typically, this practice includes providing, during or prior to replay, the same values to the memory locations that were present during the initial

## execution of the program.

Interference with the original (first executed) program often is caused by inserting instrumentation instructions into the original program to facilitate replay. If there are too many of such instrumentation instructions or if such instructions are too intrusive, then the original program may be slowed down too much during the initial execution or during the replay.

Because replay generally involves some storage during an initial execution of a program, as mentioned above, another consideration is the amount of data to be stored. Although enough data should be stored to accurately replay the desired portions of the original program execution, too much stored data will require excess storage resources as well as unnecessary overhead functions to manage the stored data.

15

20

25

30

5

10

One approach to incremental replay uses a virtual memory system of a computer to periodically trace, at fixed time intervals, the pages of the virtual memory system that were modified by a computer program since a previous checkpoint of the computer program. The term "trace," as used herein, refers to storing data in a memory so that such data is available for later use such as resetting variables and memory locations.

To restart the program's execution from an intermediate point requires that a replay tool search the stored trace to find the most recent trace of each page. Because checkpoints are taken at fixed time intervals, the system bounds replay time, i.e. the amount of time required to replay up to a desired portion of the original program execution. However, setting up the state for the replay may require searching through an entire trace file, which may involve significant time and resources. Although this approach is adaptive in that it traces only pages that have been recently written to, tracing the entire contents of pages that have been written to since the last checkpoint can require large amounts of storage.

Another approach to tracing is performed at compile-time of a program to determine what and when to trace. In particular, a prelog on the entry of each procedure is written, with the prelog containing the values of the variables that the procedure might possibly read. The prelog allows a procedure to be replayed alone since the prelog contains all variables necessary for the instructions of the procedure to properly execute. A postlog is written upon exit from the procedure, with the postlog containing the values of the variables that the procedure might have modified. The postlog allows the procedure to be skipped during replay since the postlog includes changes that the procedure might make.

5

10

20

25

30

Such a system may result in storing much more data than is actually required to facilitate replay, because the analysis performed at compile-time must be conservative to assure that any replay will be accurate. Additionally, tracing only at procedure entry and exit may incur a large amount of intrusion during the initial execution of the program, and does not guarantee that replay up to a desired portion of the execution will be attainable in a predetermined amount of time. For example, a loop that is iterated several times and includes a procedure call may result in many needless traces. Conversely, a very long procedure call may not be traced often enough to replay any part of the procedure within time constraints that are acceptable to a user.

Another system traces a "change set" before each statement or group of statements. The change set includes the values of the variables which might be modified by the statement or group of statements. A debugger then can backup execution over a statement by restoring the state from the associated change set. To bound the trace size, it is possible to store only the most recent change set for each statement. The system statically computes the change sets, and for programs that use pointers and arrays, the system must trace each such access. The system does not bound the time required to perform a replay, since it requires backing up to an instruction that is prior to the desired interval, and then progressing forward to perform the desired interval. This system is limited

4

by its static nature, in that it may have to trace every array or pointer reference for example.

Another technique maintains a hierarchy of checkpoints, each checkpoint taken at successively larger time granularities, so that recent states can be reproduced relatively quickly while older states incur more delay. This technique also includes a virtual snapshot as an approach to checkpointing, in which only the elements of a checkpoint that differ from a previous checkpoint are saved.

10 US Patent Number 5,870,607 shows a method for selective replay of computer programs. A user can selectively reply portions of a computer program execution so that the entire program need not be run again to support further test and debug. A run-time instrumented version of the program is created by inserting special instructions into the original program. The run-time instrumented version is executed to create trace files of memory accesses and system calls as well as identification of interrupts.

US Patent Number 5,642,478 shows a distributed trace data acquisition system. An event data capture circuit is integrated into each processing node in a distributed multi node system for capturing even data within each node under software control.

US Patent number 6,314,530 shows a computer system having an on-chip trace memory having a plurality of locations for storing trace information that indicates execution flow in the processor.

It is a common disadvantage of prior art tracing techniques that reviewing and modification of customising parameters is not supported.

20

25

5

#### Summary of the Invention

10

15

20

25

The present invention provides a data processing system having means for storing database tables holding customising parameters for customising an application program. In addition to the application program a trace program and an analytical program is provided.

The results provided by execution of the application program are stored and displayed for a users selection. During execution of the application program trace data is acquired by the trace program. The trace data comprises a path indication for each of the results. Each path is descriptive of a sequence of database table entries of the database tables holding a sub-set of the customising parameters used for calculation of the corresponding result.

When a user selects one of the displayed results for analysis of the underlying customising parameters that have been used for calculation of the result by the application program, the path of database table entries being assigned to the result is retrieved from the trace data. This facilitates sequential display of the database table entries identified by the path for a users review and / or modification.

In accordance with a preferred embodiment of the invention explanatory text strings are assigned to the database table entries. When a database table entry is to be displayed as it belongs to a path of the user selected result the text string assigned to that database table entry is retrieved and displayed in conjunction with the database table entry. As a consequence the displayed trace data is understandable even by a non-expert user.

In accordance with a further preferred embodiment of the invention the graphical user interface is provided that facilitates a users navigation along the path of the selected result. This way a user can go backwards and forwards along the path of the database table entries on which the calculation of the result by the application program is based. This makes the interaction of the individual database table entries transparent and facilitates a users review and /

or modification of the customising parameters stored in the respective database table entries.

The present invention is particularly advantageous as it facilitates review and modification of customising parameters on which execution of an application program is based. It is to be noted that the structure of customising information stored as customising parameters in database tables of a relational database can be of a complex nature. The present invention enables to make the structure of the customising information transparent. This is particularly advantageous for non-expert users as no understanding of the program code of the application program is required for a users comprehension of the sequence of database table entries on which the calculation of an individual result provided by the application program is based.

## **Brief Description of the Drawings**

10

In the following preferred embodiments of the invention will be described in greater detail by way of example only by making reference to the drawings in which:

- Figure 1 is a block diagram of an embodiment of a data processing system of the invention,
- Figure 2 is a flowchart illustrating acquisition of trace data,
- Figure 3 is a flowchart illustrating analysis of the trace data,
  - Figure 4 is a highly schematic output screen illustrating an embodiment of a graphical user interface.

#### **Detailed Description**

Figure 1 shows data processing system 100. Data processing system 100 has graphical user interface 102, at least one application program 104, trace program 106, and analytical tool 108. Processor 110 serves for execution of these programs. Alternatively data processing system 100 is a parallel or

distributed processing system having a plurality of processors for program execution.

Data processing system 100 has memory 112 for storing customising parameters in memory area 114, for storing results in memory area 116 for storing trace data in memory area 118, and for storing explanatory text strings in memory area 120.

In the preferred example considered here application program 104 has a number of N modules or objects that perform various program functionalities. Execution of application program 104 by processor 110 on the basis of the customising parameters stored in memory area 114 provides a number of n results that are stored in memory area 116.

10

20

25

The customising parameters are stored in memory area 114 as a relational database having a number of relational database tables T1, T2, T3, T4, ..., Ti, ...

For example database table T1 holds customising parameters P1 (T1) P2 (T1), ... Pj (T1), ... In other words, each one of the database tables Ti holds a number of parameters {Pj (Ti) }.

The database tables of the relational database stored in memory area 114 hold complex customising information. The structure of the customising information that is used for calculation of the results is described by a number of paths linking individual ones of the database table entries. By way of example a portion of one of the paths is illustrated in figure 1.

This path starts at entry P2 (T1) of database table T1 from where it goes to table entry P1 (T4) of database table T4. From there the path continues to table entry Pj (T2) of database table T2, etc. By way of example only and without restriction of generality it is assumed that this path illustrated in figure 1 holds the customising information that is used for calculation of result 1 provided by application program 104.

Memory area 120 holds a number of database tables Ti corresponding to the database tables stored in the memory area 114. In each one of the database tables Ti stored in memory area 120 an explanatory text string is assigned to each one of the parameters. In other words text string j is assigned to customising parameter Pj (Ti) of database table Ti.

For normal operation of data processing system 100 application program 104 is started without trace program 106. The results which are thus obtained are stored in memory area 116 and are displayed on monitor 122 by means of graphical user interface 102. If the user is dissatisfied by the displayed results and / or desires to analyse the basis on which the results have been calculated, he or she starts trace program 106.

10

15

20

Next application program 104 is executed one more time by processor 110 while trace data is acquired by trace program 106 and stored in memory area 118. Trace program 106 tracks which ones of the database table entries of the relational database stored in memory area 114 are accessed and in which sequence in order to calculate the n results.

The trace data consists of a table T for each one of the results. The table T (result 1) contains a list of the database table entries of the relational database stored in memory area 114 that belong to the path used for calculation of result 1. The same applies analogously to the further tables T(result 2), ...,T(result n). This way trace data is acquired and stored in memory area 118 that contains path information being descriptive of the customising parameters used for calculation of the corresponding results.

After execution of application program 104 the acquired trace data stored in memory area 118 can be analysed by means of analytical tool 108.

The user can select one of the results provided by execution of application program 104 by means of graphical user interface 102. Analytical tool 108 retrieves the table T corresponding to the selected result from memory area 118. Analytical tool 108 reads the value of the first element of the sequence,

e.g. P2 (T1) from the corresponding database table T1 stored in memory area 114 if result 1 is selected and displays the value by means of graphical user interface 102.

Further analytical tool 108 retrieves the explanatory text string belonging to the first element of the path. e.g. P2 (T1) from memory area 120. In other words text 2 assigned to P2 (T1) in table T1 stored in memory area 120 is displayed in conjunction with the value of that customising parameter P2 (T1). By means of graphical user interface 102 the user can navigate along the path belonging to the selected result. This will be explained in more detail by making reference to the example shown in figure 4.

5

10

20

Figure 2 shows a flowchart for acquisition of the trace data. In step 200 the trace program is started. Next the application program is started at step 202. In step 204 trace data is acquired during execution of the application program.

The acquired trace data includes at least path information regarding the subsets of the database table entries stored in the customising parameter memory area that have been used for calculation of the results. Hence, the acquired trace data acquisition contains lists, i.e. tables T(result), being descriptive of the database table entries that have been used for calculation of a result.

After termination of the application program (step 206) the analytical tool is started in step 208. By means of the analytical tool a user can review and / or modify the customising parameter settings that have been used as a basis for calculation of the results (step 210). The review and / or modification process of step 210 is explained in greater detail by making reference to the flowchart of figure 3.

In step 300 of the flowchart shown in figure 3 the results provided by the application program are displayed. In step 302 the user selects one of the results for further analysis. In step 304 index m is initialised. In step 306 the first element of the customising parameter path belonging to the selected result is retrieved and its value is read from the relational database. For example, if

result 1 has been selected for analysis, the first element of the trace data table T (result 1) stored in memory area 118 (cf. figure 1), i.e. parameter P2 (T1), is read and its value is retrieved from the relational database stored in memory area 114.

The explanatory text string assigned to the first element of the customising parameter path is retrieved from the corresponding table stored in memory area 120 in step 308. In step 310 the first element of the customising parameter path is displayed in conjunction with its explanatory text string. In step 312 the user has the option to modify the first element. If the user does not enter a new value for the first element, the first element remains unchanged. Otherwise the relational database stored in memory area 114 is updated with the new value of the first element i.e. P2 (T1) in the example considered here.

In step 314 the index m is incremented and the control goes back to step 306. Steps 306 to 314 are repeated until the complete customising parameter path of the trace data of the selected result has been 'scanned' for review and / or modification of the user.

15

20

Depending on the implementation the user may stop the scan along the customising parameter path at any time in order to jump backwards or forwards along the path or for selection of another result for analysis and / or modification.

Figure 4 shows a highly schematic embodiment of an interactive window 400 of analytical tool 108 provided by graphical user interface 102. Window 400 has a results section 402 for display of the n results provided by execution of application program 104 (cf. figure 1).

Further window 400 has output field 404 for outputting of the value of a customising parameter and output field 406 for outputting of the explanatory text being assigned to the customising parameter displayed in output field 404. Input field 408 serves for a users input of a new value of the customising

parameter shown in output field 404. Navigation buttons 410, 412 serve for a users navigation along a selected customising parameter path.

In operation the user starts the analytical tool 108 (cf. figure 1) after acquisition of trace data by trace program 106 during execution of application program 104. Analytical tool 108 displays the n results provided by application program 104. The user can select one of the n results by clicking on the display of the corresponding result in result section 402.

In response the value of the first customising parameter of the customising parameter path belonging to the selected result is displayed in output field 404 and the explanatory text belonging to the first customising parameter in the customising parameter path is displayed in output field 406. By clicking on navigation button 412 the user can go to the next customising parameter along the customising parameter path belonging to the selected result. Likewise the user can go backwards along the customising parameter path by clicking on navigation button 410. Depending on the implementation additional navigation buttons can be provided in order to facilitate a users navigation along the selected customising parameter path.

10

15

20

If a user desires to modify a customising parameter he or she can enter a new value for the customising parameter in input field 408. In response to pressing the enter button the previous value of the customising parameter shown in output field 404 is updated by the new value.

# List of Reference Numerals

100	Data Processing System	
102	Optical User Interface	
104	Application Program	
106	Trace Program	
108	Analytical Tool	
110	Processor	
112	Memory	
114	Memory Area	
116	Memory Area	
118	Memory Area	
120	Memory Area	
122	Monitor	
400	Window	
402	Result Section	
404	Output Field	
406	Output field	
408	Input Field	
410	Navigation Button	
412	Navigation Button	

#### CLAIMS:

- 1. A data processing system comprising:
  - means for storing database tables holding customising parameters for customising an application program,
  - means for execution of the application program,
  - means for execution of a trace program, the trace program operable to track which database table entries of the database tables are accessed and in which sequence,
  - means for execution of an analytical program,
  - means for storing a set of results provided by the application program,
  - means for storing trace data acquired by the trace program during execution of the application program, the trace data comprising a path for each of the results, each path being descriptive of a sequence of database table entries of the database tables holding a sub-set of the customising parameters used for calculation of the corresponding result,
  - means for displaying the results on a user interface, wherein the analytical program is operable to
    - (a) receive a users selection of one of the results being displayed on the user interface,
    - (b) display sequentially the database table entries identified by the path of the selected one of the results.
- 2. The data processing system of claim 1, further comprising means for storing explanatory text strings assigned to the database table entries, the analytical program being operable to display one of the text strings assigned to a displayed database table entry.
- 3. The data processing system of claim 1 or 2, further comprising graphical user interface means for a users navigation along the path.

- 4. The data processing system of claim 1, 2 or 3, the analytical program being adapted for entering of a user's modification of a displayed database table entry.
- 5. A method of displaying customising parameters of an application program, the customising parameters being stored in database tables, the method comprising:
  - execution of the application program,
  - acquisition of trace data during execution of the application program, the trace data comprising a path for each of the results provided by the application program, each path being descriptive of a sequence of database table entries of the database tables holding a sub-set of the customising parameters used for a calculation of the corresponding result,
  - displaying of the results,
  - interactively selecting one of the results for further analysis,
  - sequentially displaying the database table entries identified by the path of the selected one of the results.
- 6. The method of claim 5, further comprising the steps of:
  - retrieving an explanatory text string assigned to a database table entry to be displayed,
  - displaying the text string together with the database table entry.
- 7. The method of claim 5 or 6, further comprising navigation along the path.
- 8. The method of claim 5, 6 or 7, further comprising entering of a modification of the displayed database table entry.
- 9. A computer readable memory having instructions stored thereon for execution by a processor, the instructions comprising
  - trace program instructions for acquisition of trace data during execution of an application program, the trace data comprising a path for each of

the results provided by the application program, each path being descriptive of a sequence of database table entries of database tables holding a sub-set of customising parameters of the application program used for calculation of the corresponding result,

- display program instructions for displaying the results on a user interface,
- interface instructions for receiving a user selection of one of the results being displayed and for sequentially displaying the database table entries identified by the path of the selected one of the results.
- 10. The computer readable memory of claim 9, the instructions further comprising program instructions for retrieving of an explanatory text string assigned to a database table entry to be displayed.
- 11. The computer readable memory of claim 9 or 10, the instructions further comprising graphical user interface instructions to enable a users navigation along the path.
- 12. The computer readable memory of claim 9, 10 or 11, the instructions further comprising instructions for entering of a users modification of the displayed database table entry.

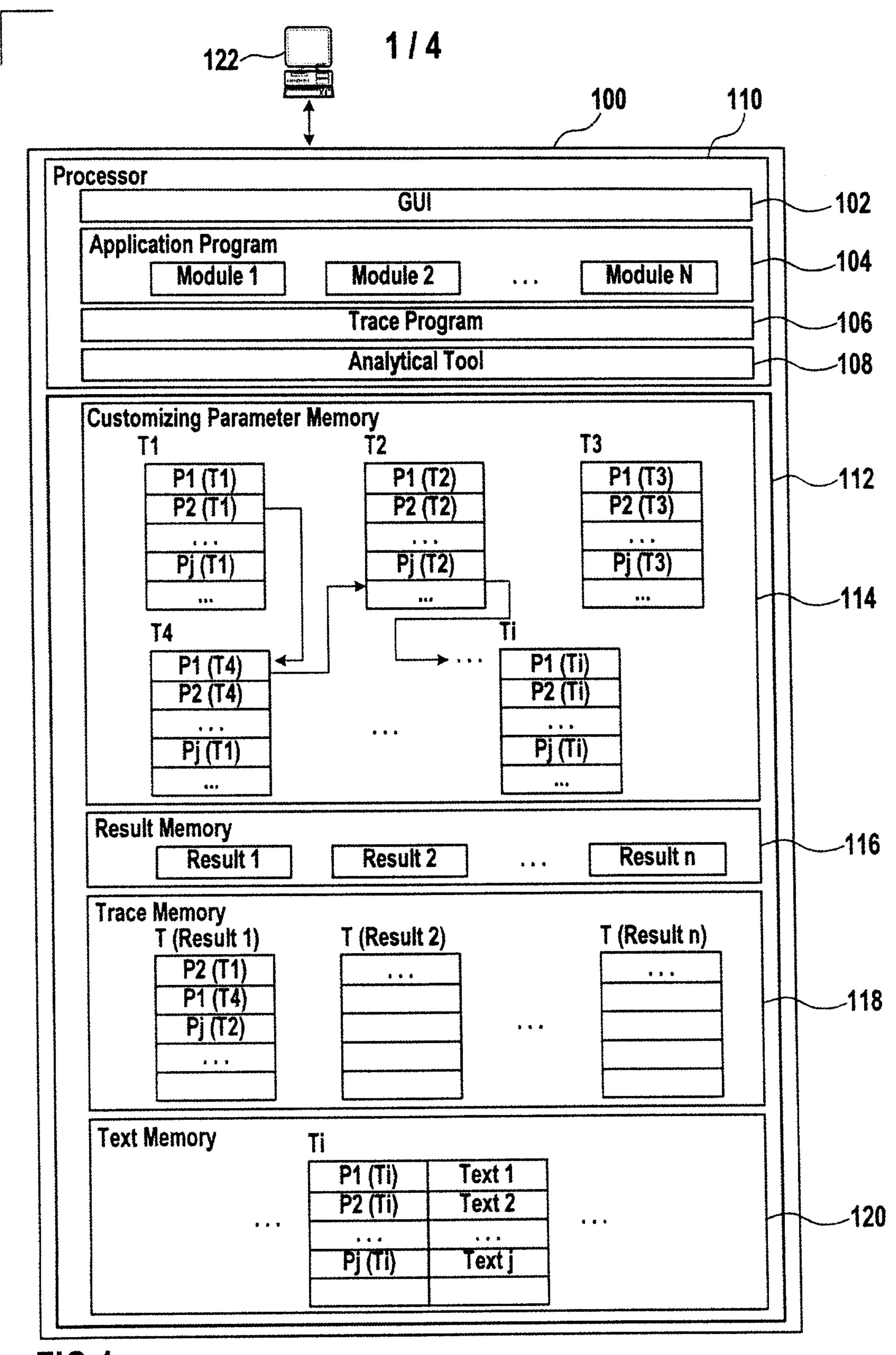


FIG.1

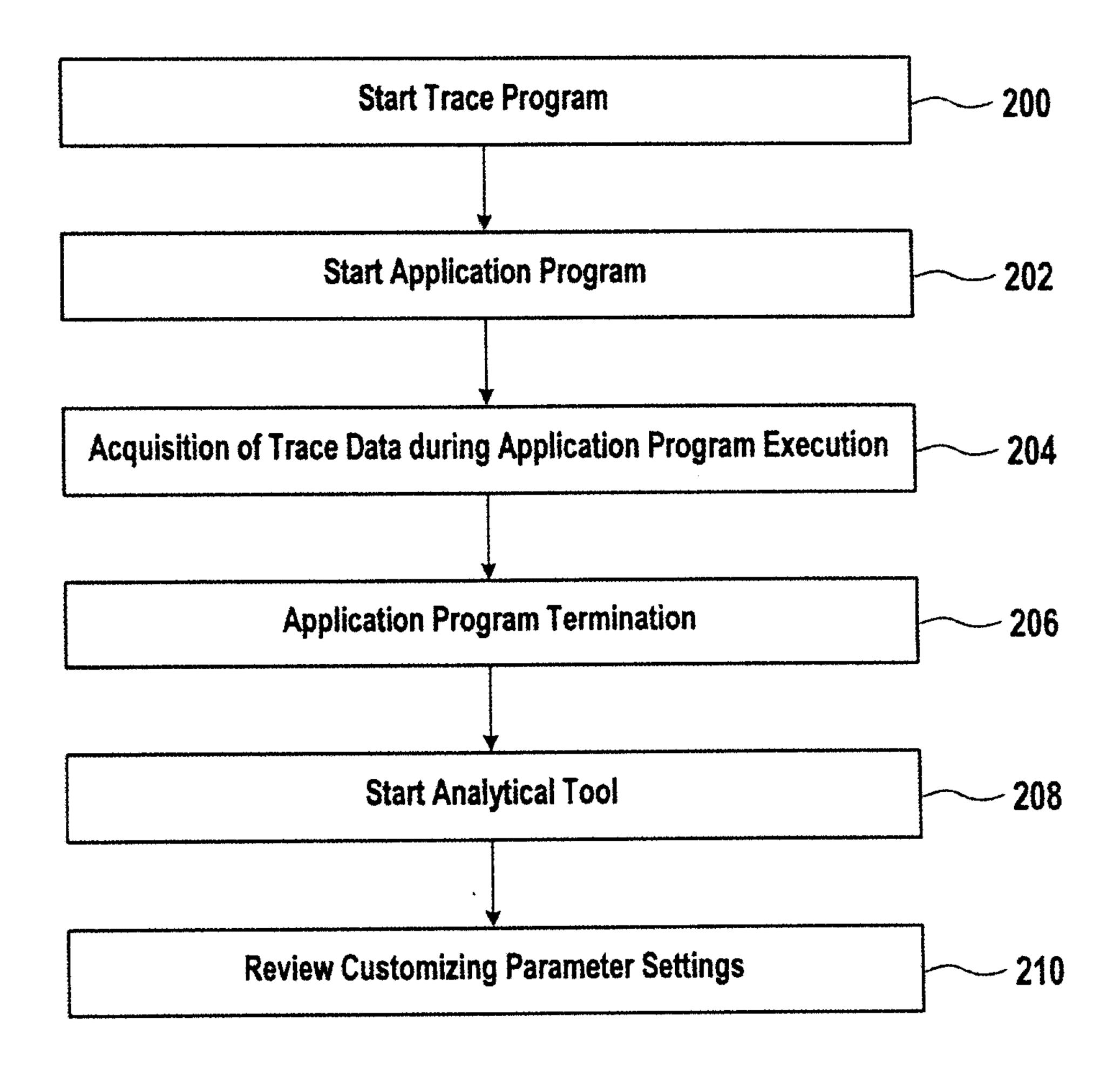


FIG.2

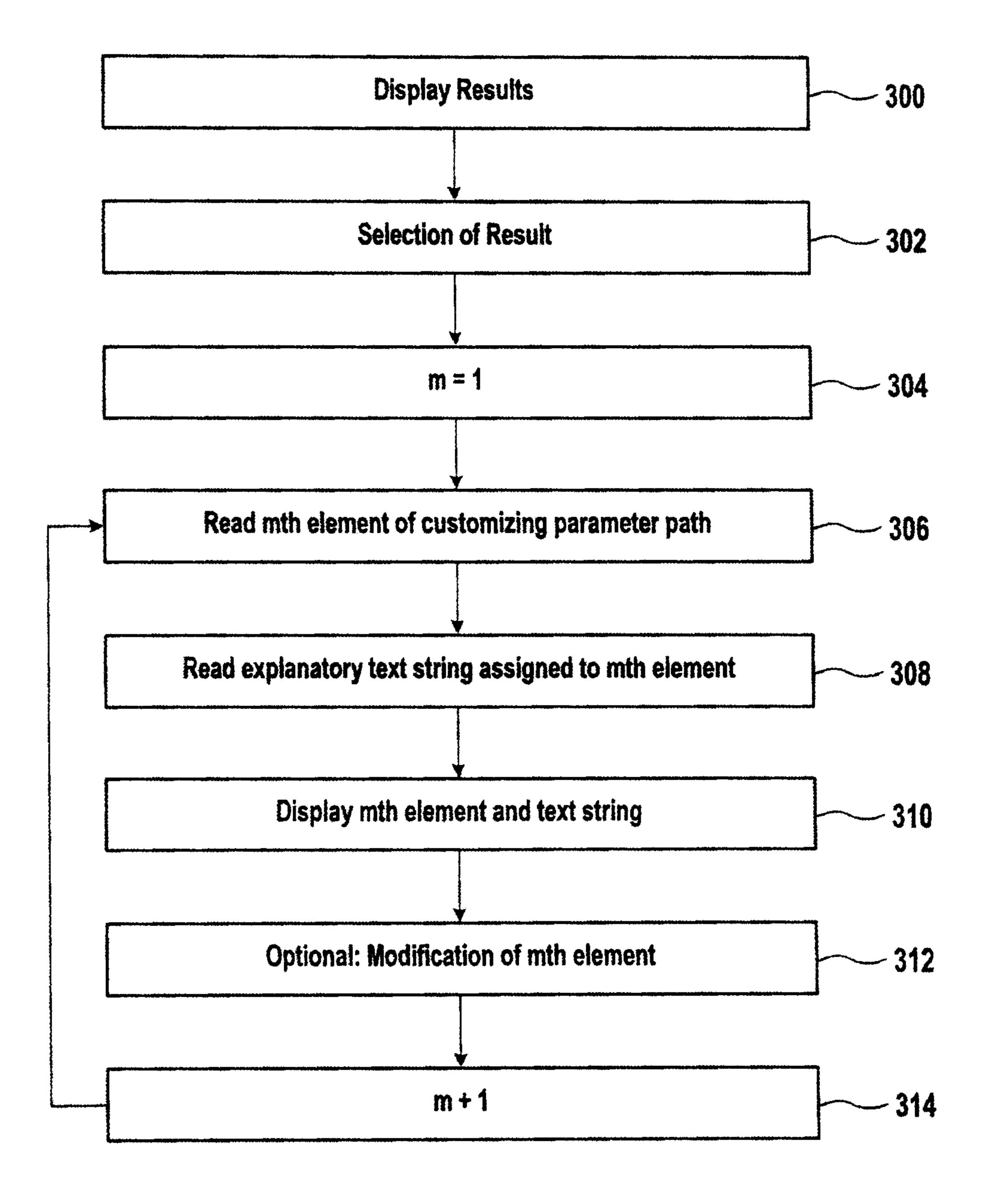


FIG.3

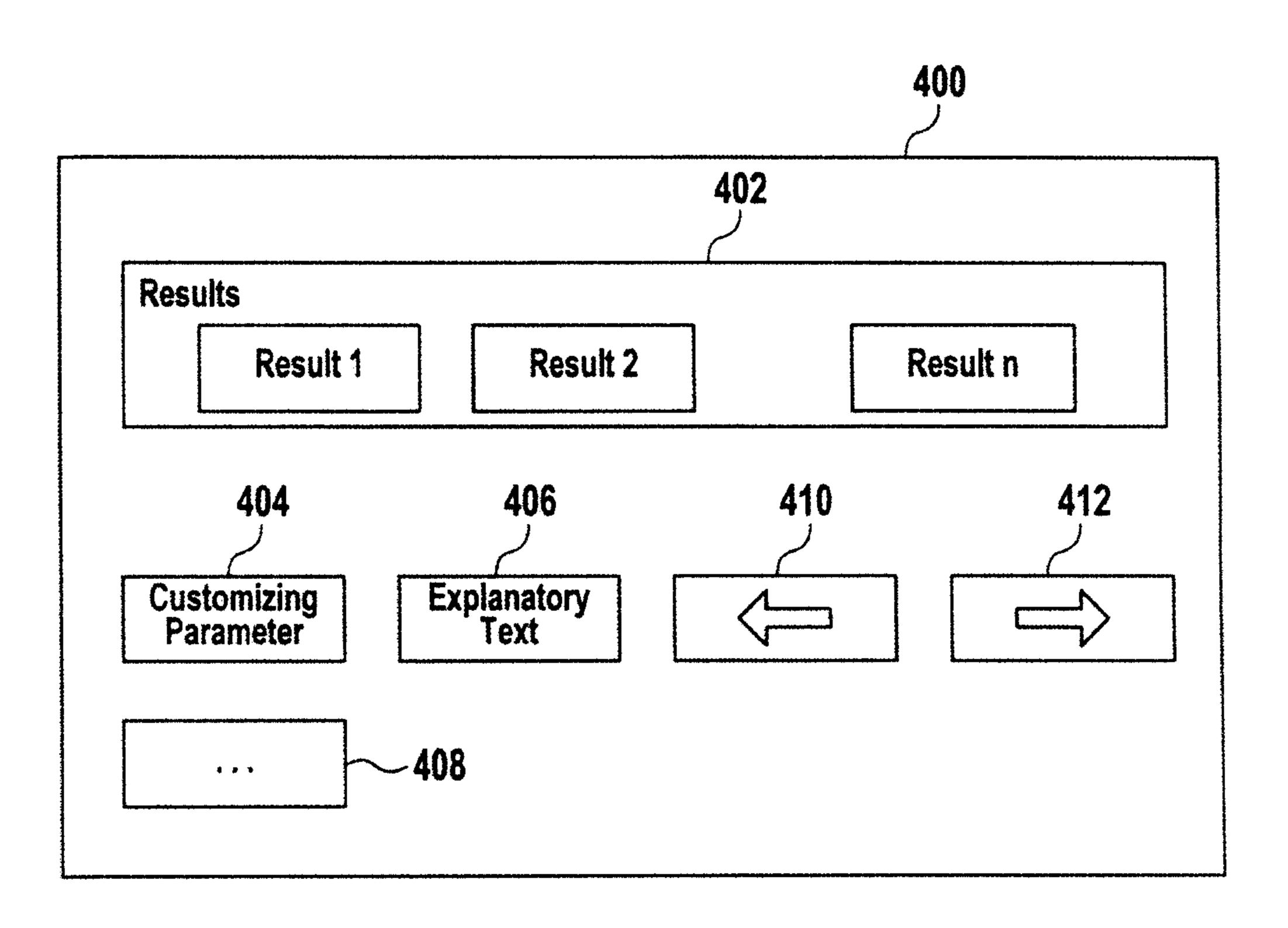


FIG.4

