

5

## **Control System for Controlling Software Modules**

The invention relates to a control system for controlling data processing software modules.

10

From prior art it is known that, at particular times of the day, a large amount of data has to be processed regularly on mainframe computer systems of large enterprises. Such data accruing regularly in large amounts will occur, for instance, when, at the end of a business day, the data of the business processes that have  
15 taken place during the day have to be processed in a so-called end of day processing.

The information accruing during the individual business processes, for instance – in the case of an industrial enterprise – who acquired something for a particular amount from whom at what time, or – in the case of a bank – who transferred a  
20 particular amount of money at a particular time or effected some other business process, is usually stored in so-called fields. All the fields containing information about a particular business process and stored in a file are usually stored in the file in a predetermined order in sequence and form a so-called data set or record.

The data sets are then collected in a file which comprises, after a certain time, the data about the business processes that have taken place since the beginning of the recording, for instance for a day or a part of a day.

- 5 The processing of such a file may, for instance, be the performance of the book-  
ing processes necessary for the business process, or the administration of a data  
stock, which has to be processed by a further system. Processing usually does  
not or not exclusively take place on the computers on which the data are created,  
i.e. the data are collected. Instead, the files are transferred to a central computer  
10 for processing, where the files created in many similar systems are collected and  
processed centrally.

Such a centralized processing involves the advantages that are usual for central-  
ized systems. With respect to the software architecture, one has recently passed  
15 over from so-called monolithic systems, in which a large program possesses the  
entire functionality required for processing and thus is capable of performing all  
the steps of processing, to a modular software architecture, so-called distributed  
systems. Here, the functionality required for processing of the data is distributed  
to a number of smaller software modules communicating with each other. The  
20 communication of the software modules with each other can be designed such  
that a first software module performs a first number of working steps for every  
data set of an input file and generates one output data set each that is stored in  
an output file. This output file of the first software module serves as input file of a  
further software module, so that one file each is generated by one software mod-

ule in a chain of such software modules and is supplied to a further one for further processing. This rudimentary kind of communication of software modules is in particular suited when, in the normal case, the processing of a large number of data sets of equal structure is performed according to a fixed scheme.

5

As compared to a monolithic system, such a modular system has a number of advantages. Thus, individual software modules of such a chain may simply be exchanged, the individual software modules are easier to change, and new modules may be introduced in the system or be replaced due to the distinctly defined  
10 interfaces between the modules.

15

Such modular systems in which a plurality of data sets accrue at a particular time in the form of such files are adapted to process the data sets within a short time. It is, in particular, often necessary to process the data sets collected in the course of  
15 a business day within a predetermined time slot so as to ensure that, for instance, particular booking processes are terminated on the same business day. An examination of the data sets for possible inconsistencies in the data sets that may be striking or may cause errors in later processing steps only cannot be performed by the first processing software module since it then would have to examine  
20 all the criteria necessary for the further processing. To this end, the entire processing would have to be reproduced in the examining software module. On the one hand, this would contradict the principle of the modular system, on the other hand, it has turned out in practice that such an examination module may also comprise errors and would have to examine an almost unmanageable amount of

possible combinations of information. Also, any change of a software module would have to be entered in the examination program. A complete examination of the data sets at the beginning of the processing is thus nearly impossible. Thus, it cannot be excluded in everyday practice that inconsistencies and errors may occur during the processing of the data in the individual software modules.

Consequently, errors and inconsistencies in the data set are possibly not recognized by the first software module in the chain of processing software modules, but only by some other, later software module, so that the error may already have caused sequential errors in other, associated systems. If a software module generates a file that is subsequently processed by several software modules, or generates several files for several post-processing software modules, the error will be multiplied. The tracing and remedying of such an error must then take place at all the affected positions of the system. The earliest possible recognition and limitation of an error to a particular data set or a group of correlated data sets is thus useful, so that the processing of a data set comprising errors may be stopped as early as possible and a propagation and spreading of the error into other systems may be prevented. For the correlation between the data sets, various criteria may be utilized. Thus, for instance, all the data sets that were collected in a particular branch office or that relate to a particular kind of business process are correlated with one another.

It is thus an object of the invention to suggest a control system that enables an early detection of data sets comprising errors or of inconsistent data sets, and that prevents the spreading of errors.

- 5 This object is achieved by a control system according to claim 1. Advantageous further developments of the invention are indicated in the subclaims.

In addition, a method for controlling data processing computer processes is provided, wherein the processing takes place in a plurality of successive processing  
10 steps and wherein the data in a processing section each are processed by at least one software module, whereby in the at least one software module of a processing section, information about the data processed is collected and signalized to a control system, and whereby the control system compares the information signalized by the software modules of the various processing sections and prevents the  
15 processing of the data in a subsequent processing section if the exceeding of a predetermined admissible maximum deviation has been detected.

Preferably, the software modules of a processing step collect, during the processing of the data sets, information about the data sets processed, for instance the  
20 number of the data sets processed that are correlated via a common feature. The information collected about the data sets processed is signalized by the software module belonging to a processing section to a control system as soon as it has processed all the input data sets fulfilling the correlation criterion. The control system thus obtains information about the data sets processed by the various soft-

ware modules of a respective processing section.. The control system then compares the information collected with respect to particular criteria in the processing software modules as to whether they concur or match with one another, or whether a predetermined admissible maximum deviation has been exceeded. If  
5 the predetermined deviation is not or just reached, the processing of the corresponding data in the software modules of the subsequent processing section is admitted, or processing in the subsequent processing section is started, respectively.

10 Otherwise, that is when the predefined additional maximum deviation has been exceeded, the processing of the corresponding data in the subsequent processing sections is not permitted, so that a propagation of the inconsistencies in the subsequent processing steps is prevented. The remedy of an error such that the reason for the deviations detected is searched, found and remedied, and the data or  
15 possibly also the software modules of a processing section are changed, is not or not automatically, respectively, performed by the present system or method. The system is rather designed to process the data delivered within the usually short period of time, so that inconsistencies occurring are not remedied in the software modules, but by interference of the operating personnel. The method thus only  
20 serves to detect inconsistencies between correlated data sets or errors, respectively, in the processing of the data sets in per se independent processing sections and to prevent their spreading in subsequent processing sections. Other errors that are, within a particular processing section, detected by this section itself

are not subject matter of the present system and method and may possibly be corrected by the processing section itself.

The data to be processed in the processing sections may also be organized such  
5 that they are stored in several files, with the information in the files being correlated with one another. The data pertaining to one business process may thus be distributed to two or more files. In a processing section, these data may be processed by different software modules of different function. These may run simultaneously on a computer system, so that the processing of the data is also per-  
10 formed simultaneously. Such a distribution of the functionality of a processing section to several simultaneously running smaller software modules instead of one large program enables, according to experience, a more favorable utilization of the computer resources, so that processing may be accelerated.

15 The control system may advantageously be further developed in that, when the predetermined admissible maximum deviation is exceeded, an optical and/or acoustic signal that can be perceived by the operating personnel is triggered, so that the operating personnel is informed of the exceeding of the admissible maximum deviations.

20

Furthermore, the method can be designed such that the further processing of the data is suppressed only for that portion of the subsequent processing section in which the inconsistency ascertained would actually lead to a subsequent error. The processing of the data may, however, be released for those subsequent

processing sections in which no subsequent error is effected. Thus, a release of the data may advantageously be performed for the processing in those portions of the subsequent processing sections for which the deviation ascertained is insignificant.

5

The method may advantageously also be further developed in that a key figure, in particular a sum, created by the bank is ascertained and signaled to the control system, and that the admissible maximum deviation is related to this key figure, in particular a sum. Such a key figure, in particular a sum, created by the bank is, for instance, the sum of all the amounts of the business processes or transactions belonging to a group of data sets. A grouping of data sets may, for instance, be performed in that the sum is created for all the data sets of a file or only for a part of the data sets, wherein, however, said data sets can be delimited against other data sets by some criterion, so that a comparison relating to a group of data sets may be performed in all the cases.

15

The admissible maximum deviation for such a key figure, in particular a sum, created by the bank may advantageously be determined as relative deviation. If, for instance, mathematical calculations of the key figures, in particular sums, are performed in a plurality of software modules, such as exchanges to other currencies, or other mathematical calculations, rounding differences may occur which may add up. It may, for example, be predetermined that the admissible maximum deviation of two key figures, in particular sums, created by the bank and relating to correlated data sets may have an admissible maximum deviation of 0.1% relating

20

to one of the two values. If the ascertained deviation of the signalized values remains below this value, the deviation is classified as an almost inevitable rounding difference and is hence not classified as an error; the data may then be released for further processing in subsequent processing sections.

5

It may, however, also be of advantage to determine the admissible maximum deviation for a key figure, in particular a sum, created by the bank as an absolute value. Rounding differences and other errors caused by mathematical operations are, according to experience, very small. Even if the absolute sums are relatively large, such errors usually occur far behind the decimal point, so that the sum of the deviations in the sum must also be small. Thus, for instance, an amount of one Euro or one Swiss Franc or of some other currency may be determined as admissible maximum deviation.

10

15

Furthermore, the number of particular processing steps performed in the software modules may be determined as a possible control value. If, for example, a number of data sets describes a particular business process in an input file of a processing section, for instance a purchase or a rebooking, and if this file is processed by two software modules, both software modules have to signalize the same number of these business processes to the control system. If, during the comparison of the respectively signalized number of business processes, a deviation is detected, there is a grave error that may have various reasons.

20

In a further variant, the system may be designed such that the time consumed by a process of a started software module to the end of the process run is taken as a control value and signalized to the control system, in particular the duration between the beginning and the end of the process. The control system then compares the time measured with a predetermined experience value and reacts appropriately.

As an experience value, for instance, a statistically adjusted mean value of the process durations from one observation period, e.g. several days and/or one or several weeks, may be used. This value may automatically be determined anew periodically by the control system.

The method may also be designed such that an input file transmitted to a processing section is divided, at the beginning of the processing in this section, into several and correspondingly smaller files pursuant to predetermined criteria. Such a division lends itself if the software architecture enables the simultaneous processing of several similar files, such as it is basically possible on mainframe computer systems of prior art. A division of a large input file into several small files then permits the processing of every smaller file by a process of a software module. According to experience, it is altogether more favorable to divide a large file into smaller ones to have these then processed by several processes of the same software module. Although the operating system and thus the computer system is loaded more strongly by the administration of the larger number of processes associated therewith, this method is advantageous and quicker than the processing

of a large file by one process since the several processes altogether make better use of the resources available in a mainframe computer system. Thus, the method may also be further developed such that the several files are processed simultaneously by every one process of a software module.

5

Furthermore, the method may also be designed such that the software modules of the processing sections are started by a permanently operating scheduler program to which an information about the success of the processing is signaled by the started processes of the software modules at the end of their run, and that the  
10 scheduler program only starts the software modules of the following processing section if the started software module signals the success of the processing and the exceeding of the admissible maximum deviation was not detected during the examination of the control values signaled to the control system. The scheduler program is used to start particular software modules at particular times of the  
15 day. The started software modules then process their respective data and report, at the end of their run, via a so-called return code, the success of the processing to the scheduler program. If a software module signals the success of the processing, the scheduler program starts the software modules of the following processing section. Otherwise, i.e. if the software module has discontinued or  
20 terminated the processing with the error status, the starting of the software modules of the following processing section is omitted. If a process of a software module is interrupted in its run due to the control values reported to the control system, this run will end in an error status. The scheduler program will then not

start the software modules of the following processing section and will thus dis-  
continue the processing of the data.

In the following, the system and method according to the invention will be de-  
5 scribed by means of three Figures.

Fig. 1 shows a simplified block diagram of the control method;

Fig. 2 shows a block diagram of such a control method by the example of  
10 a mainframe computer system of a bank; and

Fig. 3 shows a display means of the control system shown in Figure 1 and  
2.

15 The block diagram of Figure 1 shows a system 1 that generates data or collects  
the data accruing with business processes in the course of a day, respectively,  
and supplies data to the mainframe computer system 3 via the transfer channel 2  
that is illustrated schematically here. By means of a data transfer system that is  
secondary here, the data are supplied, in the form of files, to the system 3 that  
20 further processes the transmitted data. The processing of the data in the system 3  
is performed by means of the software modules 4 to 7 which are classified into  
three processing sections. The software module 4 forms the first processing sec-  
tion and supplies data that are also available in the form of files to the software  
modules 5 and 6. These, in turn, process the supplied data and deliver an input

file to be processed to the software module 7. The software module 7 forms the last processing section of the system 3 and finally enters the data in a database 9. The software modules 4 to 7 collect, during the processing, information about the processed data in accordance with particular criteria, for instance the number of the data sets processed, and communicate with the control system 8 such that they signalize the collected information – the control data – to the control system 8. This control system 8 does not process the (useful) data supplied by the system 1, but only the control data supplied by the software modules of the various processing sections and possibly by the system 1. In the instant example, it is assumed that each of the software modules 4 to 7 is to receive a data set for each data set supplied by the first system 1 to the second system 3, so that all software modules receive the same number of data sets. As soon as a software module has processed the respective input file, it supplies the number of the data sets processed altogether to the control system 8 which is thus reported the number of the respectively processed data sets from all software modules 4 to 7. The control system 8 compares, at the end of a processing section, the control values signalized by the software modules of this section and releases the processing of the data for the following processing section if the signalized numbers of the processed data sets concur or the admissible maximum deviation has not been exceeded, respectively. After the software modules 5 and 6 have completely processed the respective input files and have generated one file each for the further processing in the following processing section, namely by the software module 7, the software modules 5 and 6 signalize the respectively ascertained number of processed data sets to the software module 8. If the signalized values concur, the

files are released for further processing by the software module 7. Otherwise, i.e. if the signalized number values deviate from one another, the processing of the data is terminated and the reason for the deviation is, for instance, ascertained by means of suitable measures and is remedied. Subsequently, the processing of the corrected data is continued. During the detecting and remedying of the deviations, the system 3 is available for the processing of other data transmitted to the system. This way, the control system 8 acts like a sluice and decides, as a function of the control values signalized, whether the data that have been processed last in a processing section have to be released for further processing in the subsequent processing section.

In the following, the method will be described more concretely by means of an extract from the end of day processing of a mainframe computer system of a bank. Figure 2 shows an association of three mainframe computer systems, the boundaries of which are indicated by dashed lines. These are the so-called ABA-CUS system 10, the SSP system 11 and a third system 12 that is not designated in more detail and that receives data from the SSP system 11. Furthermore, the control system PCP (Process Control Pool) 8 is illustrated which may be operated on the same mainframe computer as the SSP 11, or on an external computer. The control system PCP 8 is a software application that may run on the SSP system 11 or on an external computer system separated therefrom, and that is operated and monitored by operating personnel.

The flow of the data to be processed between the software modules is indicated by the bold arrows in this block diagram.

The booking preparation 13 running on the ABACUS system 10 collects the data  
5 accrued on a business day. These data are collected in files. Information concerning a business process are collected in a data set that comprises, for instance, the information about the kind of the business process, the value and the branch office at which the business process has been collected first. A file composed of such data sets may contain data sets of different branch offices which they have  
10 already transmitted to the booking preparation 13.

The booking preparation 13 transfers, in the scope of an end of day processing at  
5 p.m. and 8 p.m. every day, a file to the SSP system 11 that processes the data sets further and subsequently transfers data, for instance, to the further third system 12.  
15

The SSP system 11 comprises a plurality of software modules processing the data transferred from the ABACUS system 10 or from the booking preparation 13, respectively, in a plurality of processing sections. Each processing section consists of at least one software module that is connected with the control system  
20 PCP 8 via a signalizing connection 17. Via this signalizing connection 17, the respective software module can signalize information to the control system 8 and receive information therefrom. This communication between the control system 8 and the software modules, in which the amount of data exchanged is very small,

is not performed by means of the exchange of files, but by means of operating system internal means such as so-called pipes or ports (or messages or queues, respectively) which enable a high signaling speed and are provided by every modern operating system. The control system 8 evaluates the respective information collected by the software modules about the processing performed and reacts appropriately if the predetermined admissible maximum deviations have been exceeded, namely by stopping the further processing of the data in the processes of the software modules of the subsequent processing sections and thus preventing the spreading of errors.

On the SSP system 11, the files are processed in a first processing section by a first software module, the so-called accounting feeder 14. This accounting feeder 14 performs a format conversion due to the differing sources, so that the data are subsequently available in a uniform data format that is understandable for the subsequent programs. The accounting feeder 14 performs this format conversion once for the software module booking system 15 and once for the software module business/positions 16, so that two format conversion are performed.

During the process run of the software module 14, it collects information about the file processed. This information may be collected pursuant to different criteria. It may, for instance, be the number of all bookings, a hash total about a sum to be created, which may be created depending on the currency. Likewise, the number of the different business events may be formed pursuant to the types, for in-

stance, the number of cash bookings, the stock business processes, the mere information business processes, or the pricing business processes.

In the instant example, it is assumed in the following that the information about the data sets processed is the number of the total data sets processed and the sum of the amounts of all transfers to foreign countries. This information is signalized by the software module accounting feeder 14 to the control system PCP 8.

If the process of the software module 14 does not end its run with an error message, the processing of the file supplied to the software module businesses/positions 16 may now be performed in the next processing section. This software module enters the data in the databases and generates an output file. During its run, the software module businesses/positions 16 has determined information pursuant to the same criteria as the software module of the preceding processing section, namely the number of the total data sets processed and the sum of the amounts of all transfers to foreign countries. It signalizes this information to the control system 8. The control system 8 now compares the information signalized by the software module 14 with those signalized by the software module businesses/positions 16.

If the control system 8 detects that the information about the total data sets processed or the sums deviate from one another or that the deviation of the sum values ascertained exceeds a predetermined admissible maximum deviation, the control system 8 discontinues the further processing of the data. The files gener-

ated by the software module 14 and intended for the further processing by the software module booking system 15 and also the files generated by the software module businesses/positions and intended for the further processing by the software module pricing 18 are not released for processing. This way, the spreading  
5 of an error is prevented. The control system PCP 8 is thus in the figurative sense a central supervising means having the function of a sluice that is closed for the data on exceeding of the determined admissible maximum deviations, so that the processing chain formed by the software modules is interrupted.

10 In the following, it is assumed that the control system PCP 8 has not detected any inadmissible deviation and that the processing is continued.

The output file generated by the software module businesses/positions 16 is the input file of the software module pricing 18 of the next processing section. In this  
15 software module, it is determined for each data set of the input file, i.e. for each business process, which bank fees accrue therefor respectively. The software module pricing 18 generates a corresponding output file that is intended as input file for the software module booking system 15. During its run, the software module pricing 18 has also collected information about the number of the data sets  
20 processed. It signalizes this information to the control system 8 that compares this information with the information that has been signalized to the control system PCP 8 by the software modules accounting feeder 14 and businesses/positions 16 that have run previously. If no deviation is detected or if the deviation detected

does not exceed the admissible maximum deviation, the further processing is not prevented.

The control system PCP 8 thus compares the control values signalized by the respective software modules after each further processing section. If the admissible maximum deviation is exceeded in such a comparison, this is recognized as an error. The further processing of the respective data is stopped and an avalanche-like spreading of the error is prevented.

10 The individual software modules of the processing sections are not running permanently, but are started by a so-called scheduler program, which is not shown here, only when the respective input file has been provided for processing. The scheduler program is running permanently and starts the software modules of the processing sections depending on different events. Thus, the software module  
15 accounting feeder 14 of the first processing section is, for instance, only started when the first file was transferred to the system SSP at 5 p.m. Every process of a software module or, in particular cases, the operating system of the mainframe computer, supplies an error code to the scheduler at the end of the run of the process, so that it is signalized whether the process run of the software module  
20 was successful or not. If an error code signalized to the scheduler indicates the non-planned and premature termination of a process or an error that has occurred during the processing, the scheduler does not start the software module of the next processing section since the error indicated determines prior to the start already that the following program run will not be free of errors.

A software module of a next processing section is not started by the scheduler, either, if the control system PCP 8 detects, on comparing the control values supplied, that the admissible maximum deviation has been exceeded and that the further processing thus has to be stopped. Such stopping of the processing of a file or, as the case may be, of several correlated files is performed in that the software modules of the corresponding next processing section are not started for the files concerned only. Like on every computer system capable of multitasking, of one software module that has to be available only once as an executable file, several processes processing different input files each may run simultaneously. Consequently, the processing of a particular file or of several particular files may be stopped or not be started without this affecting other processes of the same software module running at the same time, and thus without affecting the processing of other files.

Figure 3 shows, schematically and by way of example, a display means of the control system 8 illustrated in Figures 1 and 2, in particular a corresponding monitor 108.

As results schematically from Figure 3, a plurality of illustration elements or cells 21, 22, 23, 31, 32, 33, etc. arranged in corresponding columns A, B, C, etc. and lines D, E, F are displayed spreadsheet-like on the monitor 108, controlled by a control software program running on the control system 8.

Every illustration element is assigned to one of the processing sections 4, 7, etc. illustrated in Figures 1 and 2. As has already been explained above, the processing of the data takes place, with the system illustrated in Figures 1 and 2, partially in processing sections operated or running parallel to each other (with respect to  
5 time) and partially in successive processing sections (with respect to time).

The illustration elements 21, 22, 23 (or 31, 32, 33, etc.), illustrated on the monitor 108 to be each arranged in a line D, E, F, are each assigned to processing sections operated or running parallel to each other, and the illustration elements 21,  
10 31 (or 22, 32, etc.), illustrated on the monitor 108 to be each arranged in a column A, B, C, are each assigned to processing sections operated or running successively, wherein illustration elements 21, 31, (or 22, 32, etc.) that are positioned directly adjacent one below the other are each assigned to processing steps that are operated or running directly successively.

15 If the control system 8 detects for a particular processing section that it has been performed and that the above-mentioned admissible maximum deviation has not been exceeded ("no error case"), this is indicated in that the illustration element assigned to the respective processing section assumes a first predetermined color and/or lightness value (e.g. green, here indicated by a hatching running from the  
20 left top to the right bottom in the drawing).

If, contrary thereto, the control system 8 detects for a particular processing section that the above-mentioned admissible maximum deviation has been exceeded

(“inconsistency error case”) and the control system 8 stops the further processing of the data, as explained above, this is indicated in that the illustration element assigned to the respective processing section assumes a further color and/or lightness value that differs from the first predetermined color and/or lightness value (e.g. red, here indicated by a double hatching running simultaneously from the left top to the right bottom, and from the left bottom to the right top (cf. e.g. the illustration element 32) (“inconsistency alarm”)).

As has already been indicated above, the time required for processing the data or data sets, respectively, may also be determined by the software modules of the processing sections as a control value and be signaled to the control system 8, in particular the duration between the beginning and the end of the processing.

The control system 8 can – as has also been indicated already – compare the time measured with a predetermined experience value.

As an experience value, a statistically adjusted mean value of the processing times from one observation period, e.g. several days and/or one or more weeks, may be used. This mean value may automatically be determined anew periodically by the control system 8.

If the control system 8 determines for a particular processing section that an admissible maximum deviation between the time measured and the experience value has been exceeded (“time delay error case”), it may be provided that the

control system 8 does not – other than explained above – stop the further processing of the data, but indicates the time delay error case in that the illustration element assigned to the respective processing section assumes a third color and/or lightness value that differs from the above-mentioned first and further pre-determined color and/or lightness values (e.g. yellow, here indicated by a hatching running from the left bottom to the right top in the drawing (“time delay alarm”)).

By means of the control system 8 and the above-mentioned illustration elements 21, 22, 23, 31, 32, 33 displayed on the monitor 108 as explained above, a total overview of the current status situation is thus illustrated for the corresponding operating person.

**CLAIMS**

1. A control system (8) for controlling data-processing software modules (4, 5, 6, 7), wherein the data to be processed are organized in files of data sets, and wherein the processing is performed in a plurality of successive processing sections and the files in one processing section are each processed by at least one software module (4, 5, 6, 7),  
characterized in that  
said control system (8) compares information that has been collected about the data processed and has been signaled by the software modules (4, 5, 6, 7) of the various processing sections with one another or with plan values, and generates an acoustic and/or optical alarm message for an operating person if the exceeding of a predetermined admissible maximum deviation has been detected by said control system (8).
2. The control system (8) according to claims 1, characterized in that the processing is additionally performed in a plurality of, in particular more than ten or fifty, parallel processing sections.

3. The control system (8) according to any of the preceding claims, characterized in that the processing is performed in more than five or ten successive, independent processing sections.
4. The control system (8) according to claim 3, characterized in that said control system (8) comprises display means for illustrating a total overview of the alarm messages for all processing sections.