Solution for Monitoring the Flows through Web Dynpro ABAP Interfaces

Mihaela Osaci, Adela Berdie, Iulia Andrea, Iohana Varadi
Department of Electrical Engineering and Computer IT
Politehnica University of Timisoara/Faculty of Engineering Hunedoara
Hunedoara, Romania
mihaela.osaci@upt.ro, adela.berdie@yahoo.com, iulia2686@yahoo.es, iohana1901@yahoo.com

***Abstract*— The paper presents a monitoring solution for flow measurement through Web Dynpro ABAP interface. The use of integrated software systems in companies currently represents an essential condition for providing higher and more reliable access to information resources. The SAP Netweaver, a multilingual & multitasking system based on the three-tier client-server technology, is one of the mostly used integrated platforms implemented in companies.**

Keywords— monitoring flows; software integrated system; business application programming; Web Dynpro;

I. INTRODUCTION

The SAP (Software Application Products) NetWeaver Application Server is the core of the SAP software stack, which enables the development of ABAP (Advanced Business Application Programming) and/or Java applications for data storage and processing [1], [2], [3], [4]. It completely supports the J2EE (Java 2 Enterprise Edition Platform) standard. The SAP NetWeaver AS (Application Server) is also developed for Web applications. The SAP NetWeaver Application Server has three-tier architecture [1], [2], [3], [4]: the presentation level, the application level and the persistence level. The three levels communicate through appropriate interfaces: Internet Communication Manager (ICM), which makes the connection to the Internet, the Remote Call Function (RCF)**,** connection through which the external software applications can have access, or another SAP NetWeaver AS system, the Database Interface for communication with the persistence level (databases). The native programming language of the system is the ABAP language, a 4GL (Generation Language) language. Besides the programs, all the development objects required by an application are created and organized in the Application Server environment.

II. OBJECTIVES

Currently, there are various concerns regarding the real-time control of the industrial processes [5], [6], [7].
This paper presents a monitoring solution for the flow measurement through Web Dynpro ABAP interface. In a power plant, the flow is measured by flowmeters, in certain specific points, such as SARASOTA 200.

The data acquisition is carried out using the program called *GAFA* (particularly for the flow measurements), by an operator of the system. The data arrive in text files and are manually entered into a SQL (Structured Query Language) sever database, after which the data are going to be monitored by SQL queries.

The solution presented in this paper is the automatic transfer of data from the text files into the database MaxDB SAP of the NetWeaver system from a SAP job that ensures the running in the background of an executable program for reading the data files, processing the data and transferring them into the database.

When finding a blank field, the data transfer stops and the operator is announced by an e-mail to correct the fault. After solving the fault, the transfer process resumes automatically. Then, through the Web Dynpro ABAP technology, the current standard Web programming, the data may be monitored via the Web Dynpro interface, carried out using technical faceless components. The program structure is shown in Fig.1.

```
REPORT  Y_PC.
START-OF-SELECTION.
...
PERFORM read_file.
PERFORM bddate.
FORM read_file.
...
ENDFORM.   " read_file
FORM bddate.
…
if indica = 1.
PERFORM trimite_email using lt_email_initial1 lt_email_final1
'Check monitoring data flow!' 'julia2686_v@yahoo.es' .
else.
...
MODIFY ytb_pc_ai from table itab_pc.
ENDFORM.
FORM trimite_email using ltc_email_initial type YTABLE
ltc_email_final type SWUOCONTTAB
lc_subject type string
lc_email type AD_SMTPADR
raising cx_send_req_bcs cx_address_bcs cx_document_bcs.
...
ENDFORM.
```

Fig. 1. The structure of an ABAP program for background running.

The block scheme for solution is shown in Fig.2. The point (a) shows the classical solution which has been already implemented. The point (b) shows the solution with the SAP background job. The solution involves setting up a SAP background Job. The Fig.2 shows the execution of programs for reading & processing the data file and saving in the database SAP MaxDB.

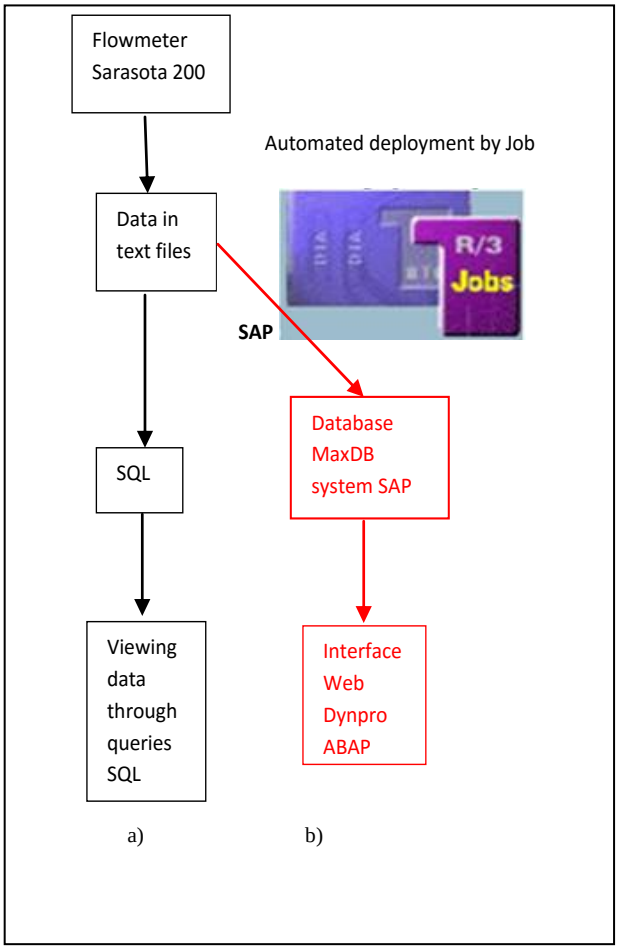

Fig. 2. Schematic Presentation of the solution for flow rates monitoring (a) The classic solution; (b) The solution suggested in this paper

III. SOLUTION OVERVIEW

The SAP background processing [1], [8] allows the automation of routine tasks and optimization of the SAP computing resources use. It provides extensive functions for scheduling and administering background jobs.

In this case, the applied transaction is SM36. The Fig. 3 shows the properties of the background Job being carried out in SAP.

In the SAP system, the *background jobs* are executed regardless of whether a user is logged in or not. When a background job is running, a batch work process is used instead of a dialog process, able to be created and regularly scheduled in all the tasks.

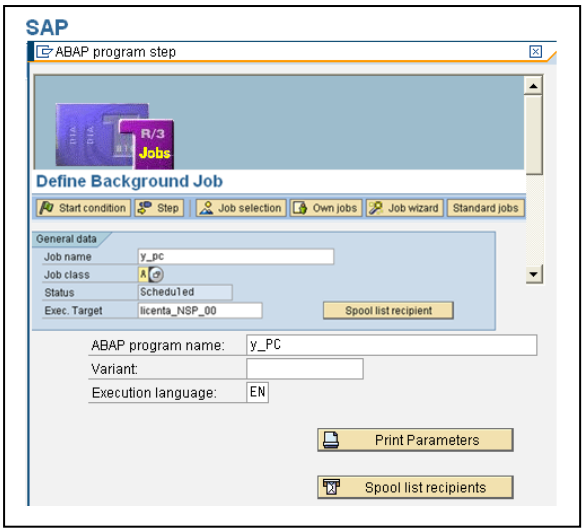

Fig. 3. Configuring Job SAP Y_PC for running in the background a program to read from the data file, to process the data and transfer them into the database SAP MaxDB.

The job execution is monitored using the transaction SM37 - Fig. 4 or transaction RZ01 - Fig. 5.

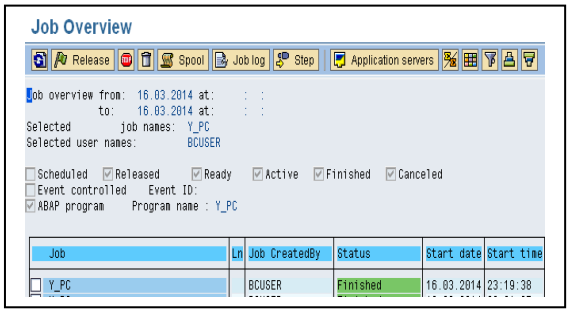

Fig. 4. The SAP background job monitoring through the SM37 transaction .

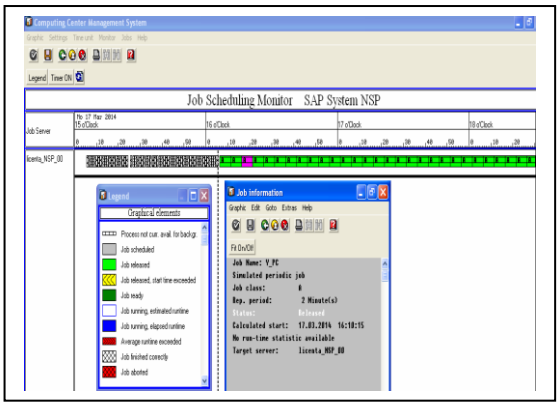

Fig. 5. The SAP background job monitoring through the RZ01 transaction.

Through this procedure, the data automatically arrive from the files into the database SAP MaxDB, in the correct format [9] – Fig. 6.

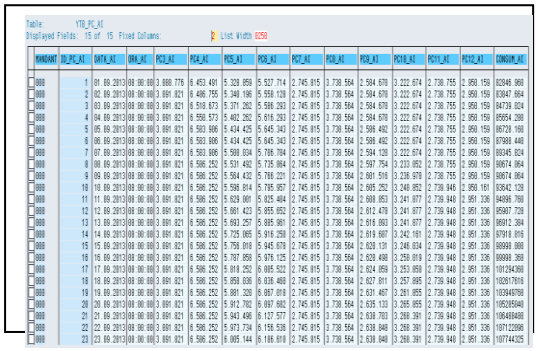

Fig. 6. Image of the database table for flow rates monitoring.

The data integrity may be verified in the program through reading, processing and monitoring of the data procedures for that purpose. For example, when a blank field is found, the data transfer process stops, and the operator is announced by an automatic e-mail to verify the data acquired on that date and time – Fig. 7.

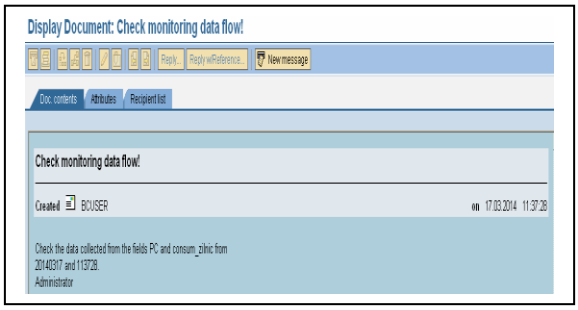

Fig. 7. Automatic e-mail notification of the operator to verify the data integrity.

After rectifying the fault, the transfer process will be resumed. Then, the data found in the database may be easily monitored via the Web Dynpro ABAP interfaces [10], [11]), designed according to the Web Dynpro technology, the current standard web programming scheme SAP [1]) – Fig. 8.

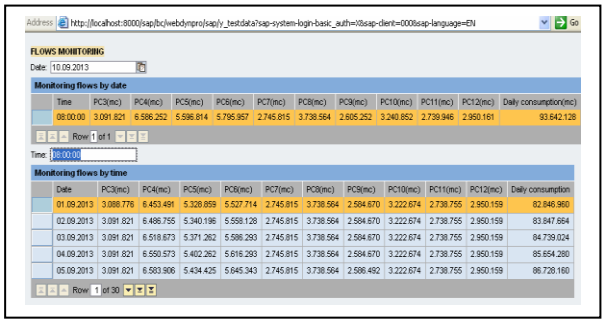

Fig. 8. The ABAP Dynpro Web interface for flow rates monitoring.

IV. THE SOLUTION PERFORMANCE IN TIME

The execution time is important information about the project performances. The execution time can be tested with the ABAP Runtime Analysis tool. The test results can be saved in graphics files on the Application Server. These files can be used to realise the necessary optimizations.

We test an ABAP program for data transfer, with and without background job. The test results are presented in Figs. 8 and 9.

From the graphics - Figs. 9 and 10, we can read, in microseconds, the execution time for the ABAP instructions, the time required to work with the database, and the operating system working time.

We can see that, in case of program with background SAP job, the total execution time is lower than in case of program without background SAP job.

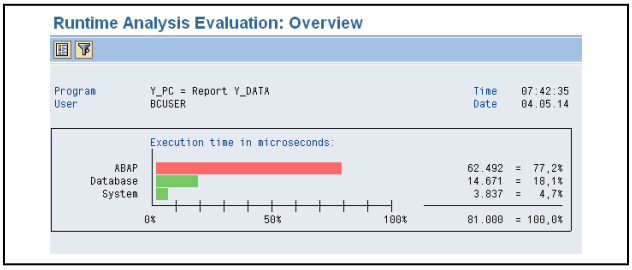

Fig. 9. Results of program testing without background SAP job.

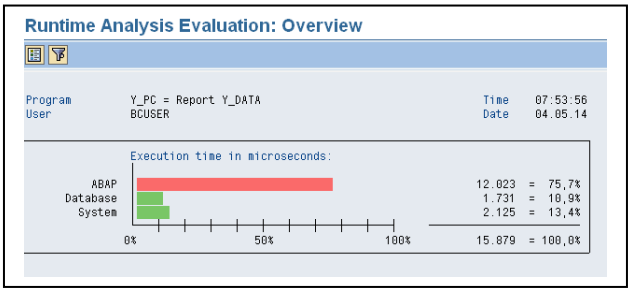

Fig. 10. Results of program testing with background SAP job.

To prevent tying up system resources with interactive sessions for ABAP programs, the SAP System has a built-in time limit on interactive sessions. If a single ABAP program runs for more than 5 minutes continuously in an interactive session, the SAP System terminates the report automatically.

The background processing system executes long-running ABAP programs more efficiently. Often, such programs are automatically scheduled for execution in the background.

V. CONCLUSIONS

This paper presents a flow rates monitoring solution by using the Dynpro Web interface, which is based on the idea of automating the process of data transfer into the database SAP Max DB, by configuring a Job for running in the background a program to read from the data file, processing the data and transferring them into the database.

The original solution requires manual data input into the database SQL Server and flows monitoring through SQL queries.

The manual data input performed by an operator requires specialized personnel, and the monitoring process is delayed

in time, and process of monitoring conventional flows requires staff specialized in the execution of SQL queries.

The solution proposed and presented in this paper solves the problem by automating data transfer from the files into the database SAP MaxDB and ensures easy monitoring flows through the Web interface Dynpro ABAP, without the need for specialized personnel to do this.

When looking for a reliable solution for the background processes, a number of companies have adopted the idea to have application servers (known as a background application server) dedicated to the background tasks.

REFERENCES

[1] SAP Help Portal (http://help.sap.com)
[2] SAP Developer Network (http://sdn.sap.com)
[3] A.D. Cristea, O. Proștean, T. Muschalik, O. Tirian, (2010) –"The advantages of using SAP NetWeaver platform to implement a multidisciplinary project", ICCC-CONTI 2010, 27th-29th May, Timişoara, Romania
[4] M. Osaci, A.D. Berdie, A.D. Hammes – "Studies on Efficientisation of the Data Access By ABAP Programming", *Soft Computing Applications*, 707-713, 2013 2013
[5] M. Panoiu, C. Panoiu, I. Sora, A. Iordan – "Study about the Process Control of an Electric Arc Furnace using Simulations based on an Adaptive Algorithm", *WSEAS Transaction on Information Science and Applications*, Issue 11, Volume 5, November 2008, 1628-1637
[6] C. Panoiu, M. Panoiu, L. Toma, R. Rob – "A Real-Time Identification Method of a Slow Process Parameters Using an Adaptive Algorithm", 8th *WSEAS International Conference on Systems Theory and Scientific Computation* (ISTASC'08) Rhodes, Greece, August 20-22, 2008, 148-153
[7] C. Panoiu, M. Panoiu, L. Tomm, R. Rob – "A Real-Time Identification Method of a Slow Process Parameters Using Adaptive IIR-OSLMS Filters", *WSEAS Transactions On Systems*, Issue 10, Volume 7, October 2008, 1143-1154
[8] http://www.datadisk.co.uk/html_docs/sap/basis/background_jobs.htm
[9] A.D. Berdie, M. Osaci, G. Prostean, R. Haipl – "Analysis and design in databases modeling using SAP integrated systems", *5th International Symposium on Applied Computational Intelligence and Informatics* (SACI'09), 247-250, 2009, Timisoara, Romania
[10] M. Osaci, A.D. Berdie, A.D. Cristea – "Studies about optimizing ABAP Codes for access to data from persistence level", *Global Journal on Technology*, no. 2, 2012
[11] A.D. Berdie, M. Osaci, I. Muscalagiu, G. Prostean – "A case-study about a web business application implemented in different SAP UI technologies", *7th IEEE International Symposium on Applied Computational Intelligence and Informatics* (SACI 2012), 111-114, 2012, Timisoara, Romania