

## B2B oriented on-line applications generator

Vintilă Bogdan-Cătălin

bogdanvi86@yahoo.com

ACADEMY OF ECONOMIC STUDIES BUCHAREST

### Abstract

B2B applications are presented. Quality characteristics of B2B applications are defined. B2B application structure is defined. The application for contracts is developed. The advantages are identified.

Key words: B2B, quality, structure, efficiency.

### 1. B2B application types

A distributed system is a collection of interdependent computers forming a computer network looks as a single computer to the users.

Middleware is a software component that connects software components or applications. It allows interaction between processes running on different computers through a network.

There are many definitions for distributed applications.

According to [SUND08] a distributed application is made of different components that run in different environments, on different platforms connected through a network.

According to [CONE08] a distributed application is an application formed by different components that interact and are physically localized on different computers from a network.

According to [IBMX08] a distributed application is defined as being composed of different components physically localized on different computers connected by a network.

A distributed application is therefore characterised by:

- it is composed by many distinct components;
- components running on different computers;
- running computers connected by a network.

Distributed applications are classified into:

- two-tier with two components out of which one is the client and the other is the server;
- three-tier formed by three components out of which one is the client, the second is the middleware and the third is the server;
- multi-tier are applications formed by many middleware and server components and one client component.

Client – server architecture separates the client from the server. This architecture is always implemented on a network. Each client is connected to at least one server. The client sends requests to the server, the server processes them and then sends the reply to the client from whom it received the request.

Three-tier and multi-tier architectures are client – server architectures for which the user interface, the functional process logic, data storage and data access are developed and maintained as independent modules, most often on different platforms.

Modularization allows independent administration, developing and modifying of modules.

Distributed applications are usable in domains where the processing power of a single computer is overwhelmed by the needed processing power, where users are distributed over a large area, or when they must access simultaneously the system, or when the same data are simultaneously used and updated.

### 2. Quality characteristics for B2B applications

*Scalability* – represents the software application’s capacity of supporting more and more users with the increase of hardware equipment. If the maximum supported number of users is reached, the increased number of computers within the network leads to augmentation of the maximum supported users without any further application modifications. If, by increasing the number of computers on which the application runs, it does not allow the increase of the supported users number, then the application is not scalable.

Applications are scalable if the indicator NrUSA –number of users supported by the application, is defined by the following relation:

$$\text{NrUSA} = \text{NrS} * \text{NrUSS}$$

where:

- NrS - number of servers within the network;
- NrUSS - number of users each server supports;

The *reliability* of the distributed application – shows the measure in which the application processes without any problems the users requests. A reliable application, assuming a computer breaks, continues its activity without any problems, recording only a drop in performance. The distributed application’s reliability depends on the number of servers on which it runs, on the distribution method, on the number of copies of the data and on resource accessing frequency.

The PNMD indicator – performance obtained provided that NMD computers of the network are broken, is defined by:

$$\text{PNMD} = \text{PM} - \text{NMD} / \text{NTMR} * \text{PM}$$

where:

- PM - maximum performance obtained when all the computers within the network are working;
- NMD - number of broken computers;
- NTMR - total number of computers within the network;

If, provided that a number of computers within the network are broken, the application’s performance is given by the above formula, then the application is reliable.

*Efficiency* – is the report between the number of useful operations and the total number of operations. The closer the value is to 1, the more efficient the application is, the closer to 0, the more inefficient the application is.

The efficiency indicator, IE, is given by the following relation:

$$\text{IE} = \text{NOU} / \text{NOR}$$

where:

- NOU - number of useful operations;
- NOR - total number of operations;

Classification of distributed applications regarding the value of the efficiency indicator

IE value	Application’s efficiency
----------	--------------------------

IE>0,75	Very efficient
0,5<IE<0,75	Efficient
0,25<IE<0,5	Somewhat efficient
IE<0,25	Inefficient

*Correctness* of the distributed application supposes a correct processing of the input data for right outputs. Data processing also supposes that it is thoroughly checked. For correct data input the application's reply will be the data processing result, and for incorrect data input the application's reply will consist in messages regarding incorrect data and their flaws.

*Maintainability* is the quality characteristic regarding the effort of modifying the application to reflect the real world changes. If the effort needed for modifying is greater than the effort required for the application's total development process, the maintaining process is suspended. The maintainable application must allow modifications with minimum effort so that, during running time the updates are not an impediment.

### 3. Applications generator

Applications generator is a software product that automatically develops applications starting from the problem's description and not through traditional programming. Based on the problem's description, using an adequate programming language, the software product creates instruction sequences and modules used in the source texts needed for solving the problem. The generator is software that develops software with a certain fixed structure. The complexity level of the created software is below the generator's complexity level. The applications generator is easier to use than the high level programming languages.

A great disadvantage is the fact that the resulting application is very slow or some of the functions can't be implemented.

Applications generators are characterised by:

- generated products have a lower complexity than the generator;
- for the same input it is always generated the same output;
- generation algorithms depend of those projecting the generator.

As users vary so do the reports that they need and the input data they supply.

The acquirement of a very complex application generates a high level of uselessness of the components leading to the idea that for every user, the distributed application's version he needs, must be generated. Even though the effort for the development of this product is greater than for developing a single application, it is by far smaller than the effort needed for developing all the applications that are afterwards generated.

If more than one application is written for solving the same problem, each of them having to be tested, the cumulated effort is greater than the effort of testing the applications generator.

### 4. Application for contracts

**The objective** of the application is to facilitate the contract's signing between the contracting parts through Internet. The need of developing a signing method that is fast, secure and easy to use is given by the increasing number of contracts companies sign as they grow stronger and stronger. Using the Internet network assures the rapidity, security and the ease of use of the application.

The application allows:

- the visualization of the transactions participants;
- signing contracts;
- the visualization of the contracts signed on an anterior date;
- signing in into the database in order to be able to use the system;

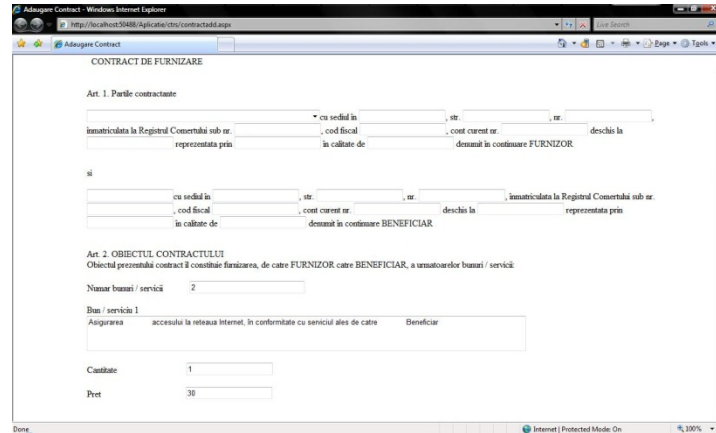


Fig. 1 - Application generates contract

**The target group** consists of companies that want to minimize the time and costs allocated to signing contracts with the partner companies. The number of users will be of thousands order and daily using.

The steps in developing the application are:

- *specification requests* supposes the description of the input data, the operations that are to be applied and the results for each request;
- *database projecting* supposes defining the tables, their structure and the relations between them, setting the domain and integrity restrictions;
- *developing the user interface* that must be friendly, intuitive and easy to use. The interface will not be overcrowded only necessary info being displayed.
- *operations implementation* supposes besides the data processing procedures, building data input validation procedures and procedures for displaying messages.
- *application testing* bases on input datasets for witch the results are already known; If differences occur between the final results from the specifications and the processed results, corrections are made and then the testing is repeated.

The contracts are generated according to the chosen parts and the user supplied parameters.

## 5. Experimental results

Testing is done by using as input the SD1, SD2, ..., datasets, for witch results exists in specifications and comparing the results obtained with those in the specifications.

Each dataset has the contracting parts, the contract's elements (goods, quantities), the prices of the contracted goods or services and the decisions that are taken during negotiations.

After using all available input test datasets and following the specified decisions, no errors were found, neither correctness errors, nor errors in the validation procedures. The generated results were totally correct.

The tests, that the application was subject to, revealed that the application meets the described quality characteristics, being an application with a high degree of correctness and maintainability.

The further development of the application will maintain or possibly improve the quality characteristics of the B2B applications.

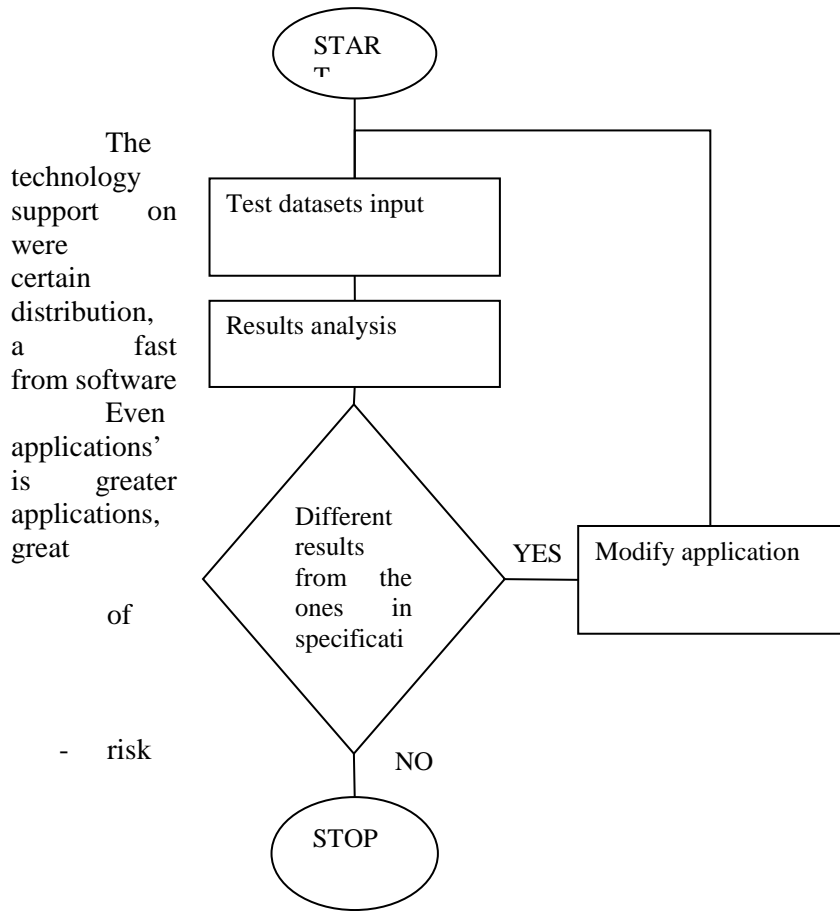


Fig. 2  
Application testing

- scalability;
- lower costs for companies;
- many users use the same data leading to saving space;
- used data is frequently updated.

Taking into consideration the great variety of applications requested by users, the generator proves its utility by generating a great part of them. The generator has many advantages:

- it allows the rapid development of applications;
- it is used by users without advanced knowledge;
- functionality is easily extendable;

In the future, the development of the generator seeks a greater variability of the results obtained through its use. For attaining this, the increase of the input parameters number is required.

As with the development of the information and communications technologies a tendency of giving up the local applications and heading towards the distributed applications is becoming visible.

## 6. Conclusions

information and communications development has been the witch distributed applications developed. As the nature of problems imposes a data or users the distributed applications meet development and great interest developers.

though the distributed development level of complexity than the one of the local the distributed applications have advantages:

- support a great number simultaneously users;
- solves problems impossible to solve by local applications;
- of losing data provided an error occurs drops significantly;
- updating distributed applications affects all the users regardless of their geographical location;
- generally the requested resources from the user side are smaller compared to the local applications;

## Bibliography

- [MACD03] Matthew MACDONALD – Microsoft® .NET Distributed Applications: Integrating XML Web Services and .NET Remoting, Editura Microsoft Press, 2003, ISBN 0-7356-1933-6, 752pg.
- [TANE95] A.S. TANENBAUM – Distributed Operating Systems, Editura Prentice Hall, New York, 1995
- [IVAN99] Ion IVAN, Paul POCATILU – Testarea Software Orientat Obiect, Editura INFOREC, București, 1999, ISBN 973-98508-0-4
- [IVAN00] Ion IVAN, Paul POCATILU, Panagiotis SINIROS – Testarea aplicațiilor e-business, SIMPEC 2000 – BRAȘOV: Lucrările Simpozionului de Științe Economice organizat de Facultatea de Științe Economice Vol. 2 , Universitatea Transilvania din Brașov, Editura Infomarket, 2000, ISBN 973-8204-00-3 ISBN 973-8204-02-X (Vol.2), pp. 172-176
- [ȘOAV00] Georgeta ȘOAVĂ – Arhitecturi client/server, SIMPEC 2000, BRAȘOV Vol. 2, Editura Infomarket, 2000, ISBN 973-8204-00-3 ISBN 973-8204-02-X (Vol.2), pp. 224-227
- [ROȘC03] Ion Gh. ROȘCA, Floarea NĂSTASE, Cristina BUCUR, Mirela VIȘEAN, Carmen TIMOFTE, Octavian PAIU – Cercetări privind proiectarea și realizarea aplicațiilor de e-business. Sisteme de plată în tranzacțiile generale de e-business, Economia, Seria Management, Editura ASE, București, 2003, ISSN 1454-0320, pp. 11-18
- [IVAN03] Ion IVAN, Marius POPA, Bogdan FLORESCU, Doru UNGUREANU – *Juridical Aspects of the Program Generators Cloning*, The 34th International Scientific Symposium – Military Equipment and Technologies Research Agency, Proceedings, vol.1, București, 29-30 mai 2003, pg 283 - 288
- [CONE08] [http://en.csharp-online.net/Glossary:Definition -  
\\_Distributed application](http://en.csharp-online.net/Glossary:Definition_-_Distributed_application), 05.04.2008
- [SUND08] <http://java.sun.com/j2ee/1.4/docs/glossary.html>,  
09.04.2008
- [IBMX08] <http://publib.boulder.ibm.com/infocenter/adiehelp/index.jsp?topic=/com.ibm.wsinted.glossary.doc/topics/glossary.html>, 09.04.2008  
[http://www.ici.ro/ici/revista/ria2008\\_1/art07.html](http://www.ici.ro/ici/revista/ria2008_1/art07.html),  
05.04.2008  
[http://www.ici.ro/ici/revista/ria2007\\_4/art03.html](http://www.ici.ro/ici/revista/ria2007_4/art03.html),  
05.04.2008  
[http://jaqm.ro/issues/volume-2,issue-1/pdfs/dumitrascu\\_popa.pdf](http://jaqm.ro/issues/volume-2,issue-1/pdfs/dumitrascu_popa.pdf), 05.04.2008  
[http://www.pcmag.com/encyclopedia\\_term/0,2542,t=application+generator&i=37909,00.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=application+generator&i=37909,00.asp), 05.04.2008