Solving Problems in Software Applications through Data Synchronization in Case of Absence of the Network

Isak Shabani[1], Betim Çiço[2] and Agni Dika[3]

[1,3] Faculty of Computer and Electrical Engineering, University of Pristina, Pristina, 10000, Kosovo

[2] Computer Engineering Department, Polytechnic University Tirana, Tirana, 1001, Albania

Abstract

In this paper, we have presented an algorithm for data synchronization based on Web Services (WS), which allows software applications to work well on both configurations "Online" and "Offline", in the absence of the network. For this purpose is in use Electronic Student Management System (ESMS) at University of Prishtina (UP) with the appropriate module. Since the use of ESMS, because of a uncertain supply of electricity, disconnecting the network and for other reasons which are not under the control of professional staff that manages the performance of this system, has interruption to the online work. In order to continue work in such conditions, are founded adequate solutions to work in offline mode and later data synchronization in normal conditions.

Keywords: *Web application, Web services, Data Synchronization, Offline mode.*

1. Introduction

With the project of digitalization of academic and administrative affairs in UP, the phase of complete computerized services has begun. For the implementation of this project from staff of Information Technology (IT), in cooperation with the management of UP the required environment of hardware and software is being prepared, respective academic network, data center and software applications are being completed. The results obtained from this work are tested and successfully implemented in the Electronic Student Management System (ESMS) also developed from UP.

The aim of this paper is to provide new results for data synchronization in different platforms through Web services, which allow software applications to run or to be executed online and offline as well [3]. The use of software applications is more productive especially when these applications can work anytime and anywhere without inconveniences related to error messages referred to the user, such as: "No access to network" or "No access to Server". In this paper, "Offline mode" development for the UP ESMS System, in order to increase confidence in using software applications is presented. However, the challenges for building and managing reliable synchronization algorithms are a major concern and potentially dangerous. Algorithm design for data synchronization through Web Services [7], used to build software applications, which besides the online mode of operation are able to work also in offline mode, in the absence of Internet and power problems, and make these interruptions unnoticed by end users, is considered [5].

2. The current state of functioning of systems in Offline mode

Offline systems so far have worked based on several techniques. Among the most advanced techniques is MS Synchronize Framework or shortly MS Sync Framework, which is a platform from Microsoft for synchronizing data from many units and offline Web applications [9]. Sync Framework can be used to access data offline. Synchronization can be used for one-to-one or one-to-many units, which function offline [11]. The synchronization service is incorporated in VS2008 and VS2010 .

This option provides synchronization of data synchronization, file synchronization and synchronization of news publications [2]. This synchronization is summarized in the following services: Sync Services for ADO.NET, Sync Services for File Systems and Sync Services for FeedSync. Thus, SyncServices for ADO.NET enables data synchronization for ADO.NET. SyncServices Service for File Systems, enables synchronization of files between two or more units with the central unit and the SyncServices for FeedSync, enables synchronization of

information in the form of RSS, Atom and distributed transaction [1]. Web applications are preferred over desktop applications because they are available 24x7 and are accessible from anywhere in the network and cloud computing [6]. A couple of years back, the term network implied wired-network but more recently, due to availability of high-speed wireless networks and handy mobile devices, the dependency over wired-network has diminished greatly. Via wireless connectivity, Web applications are now accessible literally from anywhere.

3. Data Synchronization

Optimistic replication strategies are attractive in a growing range of settings where weak consistency guarantees can be accepted in return for higher availability and the ability to update data while disconnected. These uncoordinated updates must later be synchronized (or reconciled) by automatically combining non conflict updates, while detecting and reporting conflict updates [8]. The ability to support mobile and remote workers is becoming more and more important for organizations every day. It is critical that organizations ensure users have access to the same information they have when they are in the office. In most cases, these workers will have some sort of laptop, office desktop, Smartphone, or PDA. From these devices, users may be able to access their data directly through VPN connections, Web Servers, or some other connectivity method into the corporate networks [5]. Synchronization gained great importance in modern applications and allows mobility in the context of information technology. Users are not limited to one computer any more, but can take their data with them on a laptop.

3.1 Synchronizing complex objects

Process of synchronizing complex objects will be explained following the scheme as shown in Figure 1.

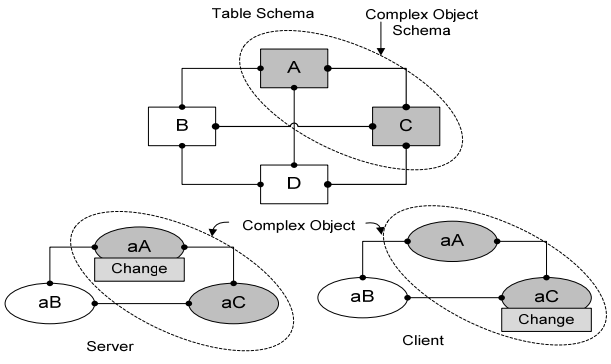

Fig. 1 Synchronization concept for complex objects .

Table A could hold zip code and city address. Table C includes street address and street number. Objects conceptually belong to each other and form a complex address of an object. Objects that conceptually belong to each other in sense of modeling in a real world are treated as a single complex object, thus the conflicts could be detected and if the data in different tables are changed. Normal process of replication would not detect the conflict and merge the data without showing them to the user. Developers can determine the objects of the class which form complex object's class. This information then is used in the time of synchronization. There are three reasons that the synchronization of complex objects is a better process: data model can be normalized and dependent data should not be considered, data model can change, however same conflicts are not detected and conflict detection in object instead of data tables is more understandable for the user.

3.2 Synchronization conflicts

A conflict can only occur when two databases have a copy of a complex object (CO) with write-permission and the complex object is changed on both sides. A complex object consists of objects that conceptually belong together in the sense of real-world modeling [4].

Complex objects CO_S in the server, contain information from different tables A, B, …, Z which are known as child-object of aA, aB, …, aZ, with aA being the root sub object. To explain the process of synchronization, let's define with:
CO_1 Copy of CO on client 1.
CO_2 Copy of CO on client 2.
t_{d1}, t_{d2} are point in time when CO_1, CO_2 are created/downloaded [4].
t_{c1}, t_{c2} point in time when sub object aI of CO_1 and sub object aJ of CO_2 are changed => CO_1' and CO_2' with $t_{c1}\neq 0$ and $t_{c2}\neq 0$ (changes occurred on both sides).
t_{s1}, t_{s2} - point in time when CO_1', CO_2' are synchronized.
t_{cs} point in time when complex object is changed on server. Precondition: $t_{d1} \leq t_{k1} \leq t_{s1}$ as well as $t_{d2} \leq t_{c2} \leq t_{s2}$ and if $t_{s1} \leq t_{s1}$, client 1 synchronizes before client 2, respectively $t_{cs}=t_{s1}$, when client 1 synchronizes, its version is saved on the server. Conflict will be detected at time t_{s2} when trying to set t_{cs} again without having seen the former version. The detection of a conflict is possible busing number ranges for unique identification. As every sub object, the root sub object of a complex object has a unique ID. CO_1 and CO_2 are copies of the same complex object if the ID of the root sub object is identical.

The allowed operations are Insert, Update and Delete. Delete is only allowed in a few tables which are checked out by the client. The conflict scenarios Update-Delete and

Delete-Delete are therefore not possible. Through the synchronization process the entries in the table are sent from client to server, executed on server side and stored in the server table. When another client downloads data it will receive the new entries from the table and can delete the records locally. Additionally, an Insert-Insert conflict is not possible either. Since every client and server has an own number range, the IDs will not be violated during synchronization. The only possible conflicting operation is Update-Update.

4. Electronic Student Management System

With the process of synchronization of data it is possible to unify the data that are held in particle databases of special physical units. When one application modifies the data in one database, relevant changes pass to other databases. In our case, we have a central database and localized database in 17 faculties. During the transmission of data, from the central database to faculties, especially after the offline work of special faculties, we should take care how we treat data movements in order to avoid potential data conflicts.

ESMS is an UP system that is developed and implemented in all faculties of UP. System is Web based, with Web application and the database in the centre, and with the Web interface for the users in all faculties. The UP has built a network that connects all faculties. In UP we still have some problems with infrastructure reconstruction. The main problem remains with the electrical power interruption, since there are still difficulties to provide with a non-stop electrical power supply all the institutions and citizens in Kosovo. The influence of power supply is a major issue to the software systems. The electrical power interruption often happens during the work of the institutions so if the officials from the specific institution is using the system, the power interruption stops the process.

This problem is of direct consequence to the officials, since they need to provide services to the students who are expecting the availability of the information all the time. This leads to a diminished confidence of the students towards the faculty. Also the officials within the institutions decrease the confidence in UP systems. Also, since the officials in some cases are not very much familiar with the usage of different computerized systems, they tend hesitate working with new systems. The electrical power interruption has its own contribution that helps in the hesitance to adapt with new technologies. The Figure 2 shows the configuration of the servers and applications in the UP IT Centre and in one faculty.

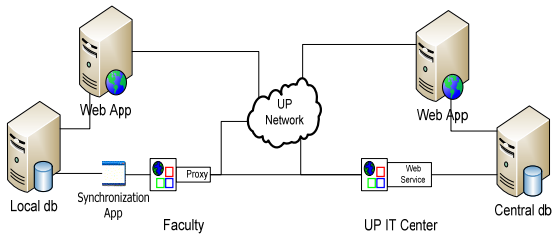

Fig. 2 Server configuration in UP IT Center and in one Faculty.

Taking into consideration these aspects and in order to succeed with this project, we had to find a solution that is independent from these problems. The solution was to introduce offline mode of operation. First thing to do in this case was to supply all faculties of UP with database and Web servers. Then, after the hardware infrastructure was completed, we had to install a copy of the application and the copy of the database in these servers. In this way, each of the faculties had the application at their disposal and could work with some of the features of the system. The problem lays still on the features of the system regarding the interconnection of other faculties and with the UP: How to synchronize the data from one side to another? We had to develop a special software component that does the data synchronization from one side to another.

4.1 Technologies used to develop ESMS

ESMS is developed using modern technologies, which are numbered in Table 1.

Table 1: Technologies used to develop ESMS

No.	*Description*	*Technology used*
1	Server Operating	Microsoft Windows Server 2008™
2	Client Operating System	Windows, Linux and mobile
3	Development Platform	.NET Framework 3.5
4	Database	MS SQL Server 2008
5	Web Server	MS IIS 7
6	Browser	Internet Explorer, Mozilla Firefox, Opera, Safari, Opera
7	Programming languages	ASP.NET, C#, AJAX , HTML, CSS, JavaScript
8	Development Tools	Microsoft Visual Studio .NET 2008
9	Accessing Data	ADO.NET
10	Communication with other app.	Web Service, XML

5. Algorithm for data synchronization in ESMS

In UP there are still problems with infrastructure reconstruction. Main problems are: non-regular power supply, connection drops, server drops, which present a big problem in software applications. Such problems cause activity interruption at work and inability to do the service on time, which reflects the service to the students. Situations of this nature; cause skepticism to students, personnel and management of UP in use of IT services to the students. If there's connection, ESMS works parallel online and offline, which means the data are transferred in both local and central databases. If the connection is lost, ESMS works with local server, which means that new data are being saved in local server which are not in the central server and in this case the synchronization component should synchronize the data with the data center when connection is present. Web Service has the information on how data should be synchronized through the columns in the tables.

Considering those aspects and for the continuity of this project, a solution to minimize the problems should be found. For this purpose an offline mode is used, realized with the design of the algorithm used for data synchronization based on Web Services, as shown in Figure 3.

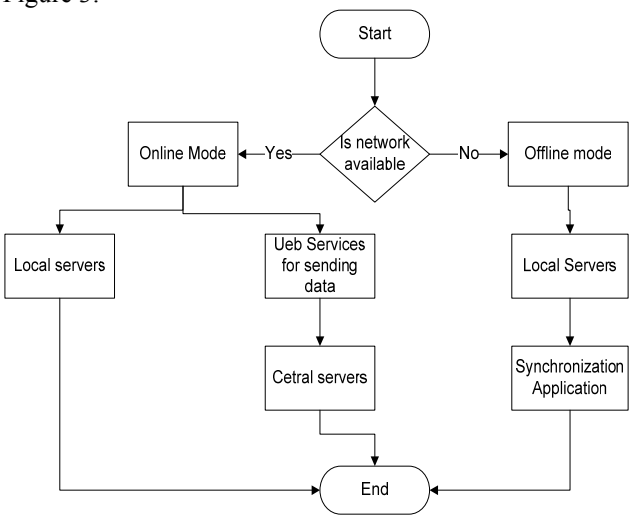

Fig. 3 Algorithm for data synchronization in SEMS .

6. Implementation of the proposed solution for ESMS in Offline mode

Implementation of Web applications in fact represents the online system, which will be set in a real Web server and connected to the Internet. Some organizational units may not have access all the time to the Web applications due to their disconnection from the network. Those units that are not included in the online Web application will be enabled during the offline mode. The offline system could be a reduced version of the entire system or an online system with rights and obligations to users. However, the data stored in the system during offline operation, in case of reactivation of the network connection will enable the data synchronization with the online system, which will regret the role as primary system.

To provide both modes of operation, ESMS has built a system which enables the data synchronization between faculties that are working offline with the primary system, in the main data base, so that the data could still be synchronous and up to date with the work done in all the different units. For this purpose, it has been provided the data synchronization, for the units working in "Offline" mode, with the primary system. This synchronization is also based on techniques that make use of WSs, considered as a relatively new technology, and the network infrastructure that will support the implementation of such technology serving the UP and its data synchronization in both "Offline" and "Online" mode and vice versa. The real database will be the same one used from the "Online" mode of ESMS. In this case a WS will have one or more methods which will be provided only from the configuration of the web server at the UP. This is because the synchronization of the units that will be working "Offline" within the UP network will not have access to other networks outside the UP.

In order to make possible the connection between the different units throughout a WS a Proxy Client will be created, such that when a certain faculty will access the University network the data synchronization will be possible. The role of the proxy client would be to call the WS methods in order to transfer all the data that has been stored Offline and to receive updated data from the real database in the case when the unit has had no opportunity to communicate with central database when the interruption of the network had occurred. Besides the data transfer from the databases that work offline, units will also transfer files to the central server, from those units that have been working offline, when the network connection will back on. Files will be transferred as data and not as files, but the central server will return the file format**.** The data synchronization of the local and central server will be done in real time. Within a specified deadline a unit will be able to determine the Proxy activation from the Windows OS and the synchronization dynamics will depend from the needs that vary from hours, days, etc. In general, data and file synchronization

is possible, but the best techniques possible in order to get the best solution have been provided from a very close cooperation with the service provider at the UP.

6.1 Data Synchronization from Client

If the network is interrupted, we switch on the offline mode solution as it is presented in pseudo-code; the data are saved only on the local database server. In this case the synchronization components start and check if the network is present. When the network becomes present, this component calls WS that sends data that has to be synchronized with the central server [10]. Below is presented the pseudo-code used for data synchronization from the client with the respective comments.

```
private void Synchronization ()
{
<WebServices> objWebService = new <WebService()>;
Synchronization Faculty >> UP_IT_Center
for (int t = 0; t < numberOfTablesForSynchronization; t++)
{
  Filling DataSet with faculty data for synchronization
}
  Sendig of data to WS, that returns two DataSets:
  DataSet for confirmation that data are synchronized
  DataSet with data that has to be synchronized from server
center in faculty
  <dsFromServerCenter[2]> = objWebService.Synchronization
(<ID>,<Data>);
  Confirmation for data synchronization
  for (int t = 0; t < numberOfConfirmedTables; t++)
        {
  Confirmation that data into respective tables in faculty db
  are synchronized with the server in the centre
  }
  Synchronization UP_IT_Center>> Faculty
  Dat Synchronization that are sent from Server in the centre
  for (int t = 0; t < numberOfConfirmedTables; t++)
 {
  Registration or modification of send data
  Filling of DataSet for confirmation of synchronized data
  }
  Sending data for confirmation in server in the centre
  objWebService.Konfirmimi(ID, <Data for confirmation>);
  End of client synchronization
}
```

6.2 Data Synchronization from Server

In the next block, we have presented the pseudo-code for dat synchronization from the server in the center.

```
[WebMethod]
 public DataSet[] Synchronization (string <ID>,
 DataSet <dsDataFromFaculty>)
{
Synchronization Faculty >> UP_IT_Center
Data synchronization that are sent from Faculty
for (int t = 0; t < numberOfTablesForSynchronization; t++)
{
  Registration or modification of send data
  Filling of DataSet for confirmation of synchronized data
 }
  Synchronization UP_IT_Center>> Faculty
  for (int t = 0; t < numberOfTablesForSynchronization; t++)
 {
  Filling DataSet with data synchronization for server in the
  centre to server in faculty
 }
 <dsDataForFaculty[0]> = < Records to be synchronized >;
  <dsDataForFaculty[1]> = < Data for confirmation >;
  Returning of the data for synchronization and confirmation
  in faculty
 return < dsDataForFaculty >;
 }
[WebMethod]
public void Confirmation(string <ID>, DataSet
<dsDataFromFaculty>)
{
 Confirmation of synchronized data in faculty
 for (int t = 0; t < numberOfConfirmedTables; t++)
{
  Confirmation of data in respective tables in db in UP IT Centre
  that are synchronized with faculty
}
}
```

7. Enrollment module and candidate acceptance in UP with ESMS

Enrollment module of candidates applying for registration in UP is implemented with purpose of avoiding manual tasks, which was practiced up to the year 2010. With this module time is reduced, cost and service quality is increased, correct and unified data are available divided for each faculty. Module for students acceptance within ESMS gives ability to see an overview of given results of candidates for the enrollment in UP, which includes: scores from preliminary (high school) studies, scores from national mature test and scores from entrance exam in faculty. Candidates can verify their personal data; receive exact information online for place, hall and time of entrance exam and final results for enrollment at all times. For the first time term students enrollment in UP in academic year 2011/12, all candidate have applied online using ESMS, from all around the world. The applicants' number has reached 17139 candidates, whilst 15734 applicants took the written test.

Statistics extracted from online applicants at all times from dbESMS database is done with the following query:

(SELECT COUNT (ApplicationID) FROM dbo.tblApplication WHERE Fshije <> 1 AND ExamID > 49 AND FakultyID = NrFac) AS 'Name of Faculty' FROM tblApplication A.

Whilst for verified statistics from ESMS at all times from dbESMS database is done with the following query:

(SELECT COUNT (VerificationID) FROM dbo.tblVerification WHERE Fshije <> 1 AND afatiID > 49 AND FakultyID = NrFac) AS 'Name of Faculty' FROM tblVerification V.

Figure 4 shows the diagram on which statistical data can be seen for numbers of applicant and those verified, from all faculties in UP, an ESMS's extraction.

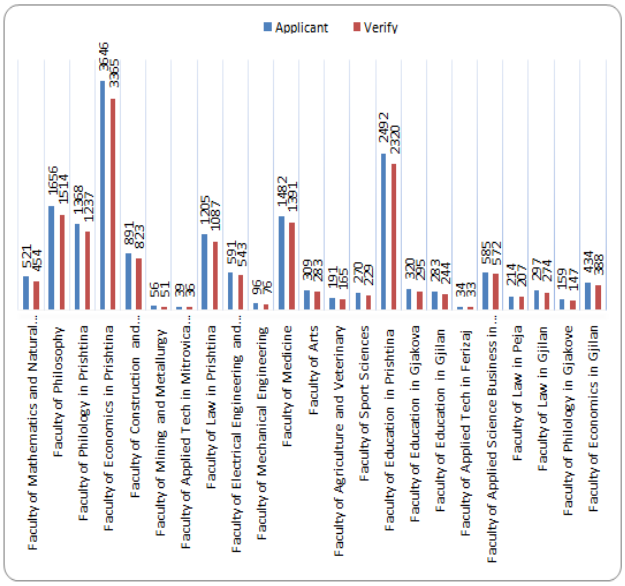

Fig. 4 Proposed beam former Online Applicants Statistics by faculty extracted with ESMS on the first acceptance term of students in UP for the academic year 2011/12.

8. Conclusions

In this paper we have studied and presented the importance of using data synchronization in an UP project called ESMS. In our study, we have investigated the WSs oriented approach for data synchronization of shared data systems, enabling software applications to work in offline mode and thus increasing the confidence in work. We used synchronization to avoid problems that the UP of Prishtina have with the interruption of electricity and the network failures. By the use of synchronization, the officials in the institutions have no more problems with the network failures, and so the system functions without any interruption. Also the confidence of the administrative and academic staff in the UP increases since there is no more waiting for the documents because the network fails. The synchronization created has shown positive results in the reliability of the users of the software applications. The synchronization that we have build for the solution of the problem on UP has found application in many electronic systems which are being developed within the e-Government in Kosovo [12]. Nowadays and in the future, system developers have to take into consideration the use of synchronization methods in order to efficiently implement their systems. In the future, application developers should consider using those methods for synchronization presented in this paper, to implement their systems with full efficiency.

Acknowledgments

Authors are grateful for all participants that helped obtained data needed to make this paper. Special thanks go to Management of UP and to the team of IT Office in UP for their contributions in the implementation, testing and administration of ESMS.

References

[1] C. George, D. Jean, K. Tim, and B. Gordon, "Distributed System, Concepts and Design", Fifth Edition, Place: published by Addison Wesley, 2011.
[2] J. N. Foster, and K. Grigoris, "Provenance and Data Synchronization", IEEE Data Engineering Bulletin, 2007, Vol. 30, pp. 13–21.
[3] K. Tomas, and S. Jakub, "Lessons Learned on Enhancing Performance of Networking Applications by IP Tunneling through Active Networks", International Journal on Advances in Internet Technology, Vol.3,No. 3 & 4, 2010, pp. 223-233.
[4] D. Dirk, and N. Christine, "A Context-Oriented Synchronization Approach", Software Competence Center Hagenberg, Softwarepark 214232 Hagenberg, Austria, PersDB, 2008, pp. 20-27.
[5] J.B. Broka, "The Future of the Internet: Scenarios and Challenges in the Evolution Path as Seen by EIFFEL Think-Tank", International Journal on Advances in Internet Technology, Vol.3,No. 3 & 4, 2010, pp. 195-202.
[6] N.S. Satish, and S. Vladimir, V. Vainikko, and J. Matthias, "Supporting Mobile Web Service Provisioning with Cloud Computing", International Journal on Advances in Internet Technology, Vol.3,No. 3 & 4, 2010, pp. 261-273.
[7] A.M. Eyhab, and H. M. Qusay, "Investigating Web Services on the World Wide Web", Department of Computing and Information Science University of Guelph, Guelph, ON, N1G 2W1, Canada , 2008.
[8] J.N. Foster, M.B. Greenwald, C. Kirkegaard, B.C Pierce, and A. Schmitt, "Exploiting Schemas in Data Synchronization", 2007.
[9] D. Lieven, J. Bart, P.Frank, and J. Wouter, "Threat modelling for web services based web applications", DistriNet Research Group, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium, 2005.

[10] P. Wohed, W.M.P Aalst , M. Dumas, and A. Hofstede, "Analysis of Web services Composition languages: The case of BPEL4WS", Proc. of 22nd International Conference on Conceptual Modeling, Chicago, IL, 2003, pp. 200-215.
[11] W.M.P. Aalst, and H.Verbeek, "Process Mining in Web Services: The WebSphere Case", IEEE Data Eng. Bull. 31(3), 2008, pp. 45-48.
[12] "E-government strategy 2009-2015", http://kk.rks-gov.net.

First Author: Isak Shabani, Dipl. Ing. In Computers and Telecommunication – 2002, Mr. sc. in Computer Engineering – 2007; PhD Candidate, Computer Engineering; Teaching Assistant at Faculty of Electrical and Computer Engineering – Department of Computer Engineering, University of Prishtina, Kosovo; has presented several papers in scientific conferences and workshops on his field of research; his current research interest include Web Services and data synchronization.

Second Author: Betim Cico, Prof. Dr. Betim Cico was graduated as electronic engineer in Polytechnic University of Tirana (PUT) in 1970 with excellent results. After one year of working as electronic engineer in Shijak Radiostation, Albanian Radio-Television, he worked for 26 years as scientific researcher at the Institute of Nuclear Physics(INP), mainly in the field of microprocessors and computers, computer based on-line systems, Head of the Electronics Department, part time professor in PUT. In 1998 he moved as Head of Computer Engineering Section Electronic Department, Faculty of Electrical Engineering at PUT. During 1999-2011 member of the main governing body (Senate) at PUT. Participated in 50 International Trainings, Workshops, Conferences etc. Author of 40 Scientific Papers and 20 INP Technical Reports. Supervisor of three PhD Theses. During 1999-2001 member of the Project Group, MOES, for the implementation of the Education Management Information System (EMIS) Component under the Transition Education Reform Project In Albania. Designer of several LANs in Tirana.

Third Author: Agni Dika, PhD in Computer Science – 1989; Full professor at Faculty of Electrical and Computer Engineering – Department of Computer Engineering, University of Prishtina, Kosovo; his current research interest include computer logic design and algorithms.