Theoretical and Applied Economics
Volume XVIII (2011), No. 8(561), pp. 31-52

User Types in Online Applications

Ion IVAN
Bucharest Academy of Economic Studies
ionivan@ase.ro
Dragoş PALAGHITA
Bucharest Academy of Economic Studies
dragos.palaghita@ie.ase.ro
Sorin VINTURIS
Bucharest Academy of Economic Studies
sorinvintturis@ie.ase.ro

Abstract. *Online applications are presented in the context of information society. Online applications characteristics are analyzed. Quality characteristics are presented in relation to online applications users. Types of users for AVIO application are presented. Use cases for AVIO application are identified. The limitations of AVIO application are defined. Types of users in online applications are identified. The three-dimensional matrix of access to the online application resources is built. The user type-oriented database is structured. Access management of the fields related to the database tables is analyzed. The classification of online applications users is done.*

Keywords: online application; user; access; metric; security.

JEL Code: C88.
REL Code: 10J.

Online applications

In the literature (Roşca et al., 2006) there are presented the coordinates of the information society which consist in:

- information creation by passing from fixed assets to informational assets, through the development of computer systems, development of digital libraries, by developing of informational portals;
- information distribution is achieved through the development of computer networks and Internet; information access is much faster and more efficient due to online search engines;
- information dissemination via the Internet, media or email;
- information usage whenever needed using personal computers or public access terminals to solve citizens problems or to improve business processes;
- information integration in complex information management systems which allow easy information retrieval using search keys;
- information management by optimizing data access processes providing for users simple and effective ways of information access.

The informatics application is a software product developed in order to be operated on a computer and to serve solving complex problems.

Distributed system is represented by a number of independent computers that communicate via a network. A distributed system is designed to solve a single problem common to all processing units or to resolve a number of issues specific to each processing unit and the role of the distributed system is to manage the resources associated to the processing units. A distributed system has the following properties:

- fault-tolerance is the degree to which the distributed system retains its functions in case of any hardware failure of its component entities;
- network topology is the pattern of computers and peripherals interconnection that make up the distributed system;
- independence degree is reflected by the extent to which computers that form the distributed system use or are aware of other computers in the distributed system.

The access to the Internet and the development of computer networks have led to the development of distributed applications such as:

- electronic payments that are done by providers or recipients, reducing the formalities and the required time for physical payments;
- digital maps that allow routes establishment, distances calculation and viewing of satellite images (Cotfas, 2009a, pp. 466-471, 2009b, pp. 31-34);

- image sorting using color palettes for identifying the images that containing certain shades or shades combinations;
- e-government that enable an effective collaboration between state agencies and citizens through the implementation of online platforms for taxes payment or the management of state and citizen problems and responsibilities;
- orthogonality analysis of the organization identifiers to ensure registration of organizations with names significant in relation to organization identifiers registered in the database.

User-oriented online applications have the following advantages, according to (Ivan et al., 2009a, Ivan et al., 2009b, pp. 139-145):

- give access to the desired resources through online databases that store information of interest to citizens;
- reduce waiting times for solving problems or for operations execution desired by the user;
- increase the efficiency of operations performed by rapid processing of the required operations and delivering results in a much shorter time;
- achieve the link between customers and suppliers by providing a collaborative environment for problems solving, services, procurement of services, provision of goods and their acquisition;
- improve companies efficiency by increasing sales and providing access to a much greater range of customers locally and internationally;
- give citizens access to a much greater range of products in an online space where the price quality ratio is high;
- make available to the public financial management systems to record individual income and expenses by eliminating the risk of mistakes and omissions made in calculations;
- citizens have access to online banking systems that allow checking account, online payments, management of bank deposits and transfers management.

Online applications are highly diverse offering varied content and the possibility to perform complex operations according to (Vintilă, Pavel, 2010, pp. 64-72).

Online applications features in regard to the user

When developing computer applications quality level is planned. According to (DEX, 1998) quality represents all the essential characteristics of an object that differentiate it from other objects. Palaghita (2009, pp. 38-58) defined a system of quality characteristics associated with computer applications. Online applications users take into account only certain quality characteristics that directly concern them.

Complexity represents the amount of resources for developing, testing, implementation, modification, correction and use of the computer application. McCabbe complexity is defined as:

$$C = m - n + 2,$$

where:

m - number of arcs in the graph;

n - number of nodes in the graph.

McCabbe complexity is has a maximum level when any node is referred by another one. Figure 1 shows the maximum complexity associated graph for the organizational names validation class with five interconnected nodes from AVIO software product.

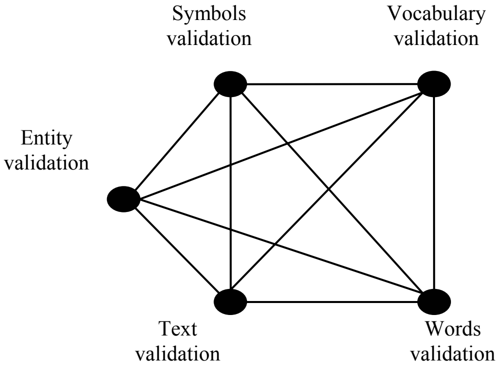

Figure 1. *The graph for AVIO names validation class*

McCabbe complexity for organization names validation class is:

$C_{name} = 11 - 5 + 2 = 8.$

Maximum complexity of a module must not exceed the value of 10 in order to ensure appropriate testing process, to ensure product reliability and maintain simplicity of the software design. Minimum complexity is achieved in linear structures of applications. Figure 2 shows the linear structure for the management class of the RGB format used for comparing the organization logos in AVIO application.

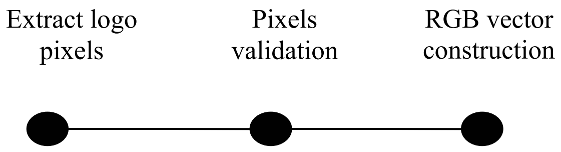

Figure 1. *RGB management class in AVIO application*

The complexity of the RGB management class in AVIO application is:

$C_{rgb} = 2-3+2 = 1.$

There are also intermediate structures combining linear structure with that of the graph type resulting complex software structures, but with an index of complexity suitable for maintaining the reliability and testability at a high level.

Accuracy according to Ivan and Boja (2004) is the degree to which results obtained from using the application are as close to real ones. The *C* accuracy is calculated using the following indicator:

$$C = \frac{N_C}{N_T},$$

where:

N_c – number of correct results;

N_T – number of total runs.

In AVIO application after running a set of tests containing 4096 of generated graphical identification elements, the accuracy index was determined with $N_e = 3072$ and $N_T = 4096$ with the value:

$$C = \frac{3072}{4096} = 0.75.$$

The correction of methods for images scaling and normalization by eliminating the default reformatting of graphic identifiers that caused the C index value of 0.75 led to a value of the C index of 1 for the executed test set.

If a distributed system component is made up of n sub-components $\{C_1, C_2, \ldots, C_n\}$ then the accuracy level is calculated using the C_c indicator:

$$C_C = \sum_{i=1}^{n} \frac{N_{ci}}{N_{Ti}} = \sum_{i=1}^{n} C_i$$

where:

N_{ci} – number of correct results of *i* component;

N_{Ti} – number of total results of *i* component;
C_i – accuracy level of *i* component.

After running the test set were identified the values of associated variables for the software components that make up the AVIO application in order to calculate the C_c indicator according to Table 1.

Table 1

The accuracy indicator - Cc

Component	N_{ci}	N_{Ti}	C_i
Images	4,096	4,096	1
Company	4,096	4,096	1
NameArray<T>	4,096	4,096	1
ComplexArray	4,096	4,096	1
Complex	4,096	4,096	1
ImageMetrics	3,072	4,096	0.75
BitmapAlredyLoaded	4,096	4,096	1
Validator	4,096	4,096	1
ImageValidator	4,096	4,096	1
UnmanagedImage	4,096	4,096	1
RGBL	4,096	4,096	1
RGB	4,096	4,096	1
Histogram	4,096	4,096	1
ColorSetLocations	4,096	4,096	1
ColorLocationList	4,096	4,096	1
ColorLocation	4,096	4,096	1
BmpStatisticsHelper	4,096	4,096	1
BmpHelper	3,072	4,096	0.75
StatisticsHelper	4,096	4,096	1
LogHelper	4,096	4,096	1
PairColours	4,096	4,096	1
LogWriter	4,096	4,096	1
Cc			0.97727273

The test set shows a value of the C_c identifier associated to the AVIO application before resolving the defects identified during testing of 0.977. After resolving the defects the C_c index value increased to 1.

Continuity is the quality characteristic represented by the degree to which changes in the interface of the software versions occurs. Considering the set of software versions SV = $\{V_1, V_2, \ldots, V_k\}$ which has an associated set of versions of the software interface SI = $\{I_1, I_2, \ldots, I_k\}$. A software interface version I_i associated to the software version V_i is defined by a set of user visible components $SCI_i= \{C_{i1}, C_{i2},\ldots, C_{im}\}$. A high degree of continuity for an IT application is defined by a high degree of similarity between the user interfaces associated with each version. This requires a high degree of similarity between components sets that make up the interfaces. Changes applied to the interface components are:

- *interchange* is defined by changing positions of two components of the interface between them;
- *exclusion* represented by deleting certain interface components that are not representative for the current version of the software interface;
- *addition* is the operation through which is created a new interface component that is included in the components set associated with the last interface version of the software product;
- *change* which alters the shape, length or content of user interface components of the software application.

Continuity of user interface I_i is defined by the CT indicator:

$$CT_i = \frac{\sum_{j=1}^{m} \frac{\min(TC_{ij}, CN_{ij})}{\max(TC_{ij}, CN_{ij})}}{m}$$

where:

TC – total interface components;

CN- number of unmodified components in regard to the I_{i-1} interface.

The CT_i aggregate indicator of continuity is easily determined by analyzing the dynamics of interface components and is a useful metric in determining the fluctuations occurring in software interfaces.

Security is the quality characteristic of a distributed system that is achieved through the ability to protect the logical and physical resources of the system. Security[(1)] is represented by the measures taken to protect a system. Security is also considered as a condition of a system which results from the establishment and preservation of some measures to protect the system. Security is a condition of system resources to which unauthorized access, unauthorized or accidental changes, destruction or loss are not allowed.

Security of a system is influenced by the used network communication protocols, network topology, user authentication methods, and used encryption systems and by the human factor involved in all aspects of the distributed system life cycle. Security is represented by the access control to software and hardware resources and to the application components. Computer systems implement the following methods to secure the access to resources:

- *authentication* is the process of identity validation; before an application authorize the access to protected resources the authentication processes are necessary for establishing user's identity and to verify that the information passed in the authentication process are recorded in the associated users group database of the distributed system;
- *authorization* is the process of setting the user access permission to protected resources within the distributed system; even if a user has proved to be recorded in the users group database it doesn't mean that it has access to the system's resources;
- *data protection* is the process of ensuring confidentiality and integrity of data stored in distributed system databases; encryption provides data confidentiality; data integrity is ensured by using digital signatures, hash algorithms and message authentication codes.

Security is an important issue for online applications according to (Doinea et al., 2010) being necessary to preserve the integrity of applications and as well as data for imposing confidentiality level.

User access restrictions in AVIO application

The application for organizational identifiers analysis determines the level of identifiers orthogonality in order to eliminate the situations where there are two companies with very similar names and logos.

In the module for organization names orthogonality analysis is built the vocabulary VOCDEN = {D_1, D_2, …, D_k} that contains all names entered in the organizations database. It is considered that the entity has a name consisting of several words and orthogonality is determined by analysis at the vocabulary level and at the word level. Vocabulary level analysis is done first in order to determine the correspondence between texts and for the requirements of application validation for orthogonality calculation.

In order to examine the orthogonality of two texts T_1 and T_2 two vocabularies are built, V_1 and V_2, that are defined by alphabetical sorting of the words that compose the T_1 and T_2 texts. Vocabularies are defined as sorted sets of words V_1 = {C_{11}, C_{12},..., C_{1n}} and V_2 = {C_{21}, C_{22}, ..., C_{2n}} where C_{1i} corresponds to the word from the *i* position of the V_1 vocabulary and C_{2j} corresponds to the word from the *j* position of the V_2 vocabulary. Names orthogonality is represented by the following formula:

$$ORTOT(V_1, V_2) = 1 - \frac{NCC}{\max(NoCV_1, NoCV_2)}$$

where:

NCC – number pf common words;
$NoCV_1$ – number of words from the V_1 vocabulary;
$NoCV_2$ – number of words from the V_2 vocabulary.

The orthogonality analysis of the organization logos is done by the compared analysis of the differences between two logos and it results an orthogonality index. A logo is a unique graphic used by organizations, companies or individuals to be publicly identified. A logo has the following characteristics:

- uniqueness, thus each logo is associated with one entity facilitating the identity recognition of the owner;
- standardization, which is reflected in the logo format in a way to ensure the compliance with other existing logos format; is aimed the width, height and quality of the logo;
- representation, which means the degree to which the logo is representative for the defined scope of the owner;
- simplicity, which is characterized by a representation which is easy to remember having a complexity of the alphabetical and graphical elements as light as possible;
- impact, which is the characteristic that determines the success of the logo considering the design elements influencing public opinion on the entity that is represented by the logo;
- color; exists logos that use only a limited set of colors that are considered representative of the entity, and logos using only black and white;
- shape differentiates logos depending on the geometric elements like round, square, oval, rectangular, hexagonal or logos composed of composed geometric shapes; as a symbol contains more geometric elements its complexity increases and the degree in which it is easily associated with the represented entity decreases.

Two logos are considered different if:

- there are different color elements between the two logos, so there is a chromatic difference in the comparative analysis;
- there are different geometry elements composing the logos so that their visual comparisons reveal obvious differences.

To determine the orthogonality for the S_{beta} and S_{beta2} logos in AVIO application in order to validate the organization identifiers, it is determined the differences between the two logos as follows:

- determining the matrices of pixels associated to the S_{beta} and S_{beta2} logos scaled to the following sizes:

- 4x4 pixels, creating the matrix $MS_{beta4x4}$ associated to the S_{beta} logo and the $MS_{beta24x4}$ matix associated to the S_{beta2} logo; each matrix contains four columns and four lines of pixels;
- 8x8 pixels, creating the matrix $MS_{beta8x8}$ associated to the S_{beta} logo and the $MS_{beta28x8}$ matix associated to the S_{beta2} logo; each matrix contains eight columns and eight lines of pixels;
- 16x16 pixels, creating the matrix $MS_{beta16x16}$ associated to the S_{beta} logo and the $MS_{beta216x16}$ matix associated to the S_{beta2} logo; each matrix contains sixteen columns and sixteen lines of pixels;

- are created two sets of matrices ST_{beta}= {$MS_{beta4x4}$, $MS_{beta8x8}$, $MS_{beta16x16}$} and ST_{beta2} = {$MS_{beta24x4}$, $MS_{beta28x8}$, $MS_{beta216x16}$};
- is realized the comparison of the sets of matrices by using the following weighted formula:

$$\mathrm{ORTOST}(\mathrm{ST}_{\mathrm{beta}}, \mathrm{ST}_{\mathrm{beta}2}) = \frac{(\mathrm{ND}4 \times 16 + \mathrm{ND}8 \times 4 + \mathrm{ND}16)}{256 \times 3}$$

where:

ND4 – number of registered differences between the $MS_{beta4x4}$ and $MS_{beta24x4}$ matrices; maximum number of differences is 4 × 4 = 16;

ND8 – number of registered differences between the $MS_{beta8x8}$ and $MS_{beta28x8}$ matrices; maximum number of differences is 8 × 8 = 64;

ND16 – number of registered differences between the $MS_{beta16x16}$ and $MS_{beta216x16}$ matrices; maximum number of differences is 16 × 16 = 256;

$ORTOST(ST_{beta}, ST_{beta2}) \in [0,1]$ because if there are a maximum number of differences between the two sets of matrices, meaning ND4 = 16, ND8 = 64 and ND16 = 256 then:

$$\mathrm{ORTOST}(\mathrm{ST}_{\mathrm{beta}}, \mathrm{ST}_{\mathrm{beta}2}) = \frac{16 \times 16 + 64 \times 4 + 256}{256 \times 3} = \frac{256 + 256 + 256}{256 \times 3} = 1$$

In the context of working with large volumes of data as (Doinea, Pavel, 2010, pp. 72-85) for AVIO application are identified the following types of users:

- users that have access to the functionality of displaying the organizations with orthogonal identifiers stored in the database; this type of users does not requires registration to access this functionality;
- users that have access to the organization identifiers validation functionality by introducing new identifiers to determine orthogonality; this type of user requires registration to get necessary permission to use the AVIO application;

- users that have access to the management functionality of AVIO product; this type of user is unique, being represented by the administrator of the IT application that manages the submitted content and the good functioning of the software.

To determine the AVIO application flows it is necessary to determine the available terms of use for the described types of users. It is also important the interaction with application and between users types. AVIO application flows are determined based on the use case described in Figure 3.

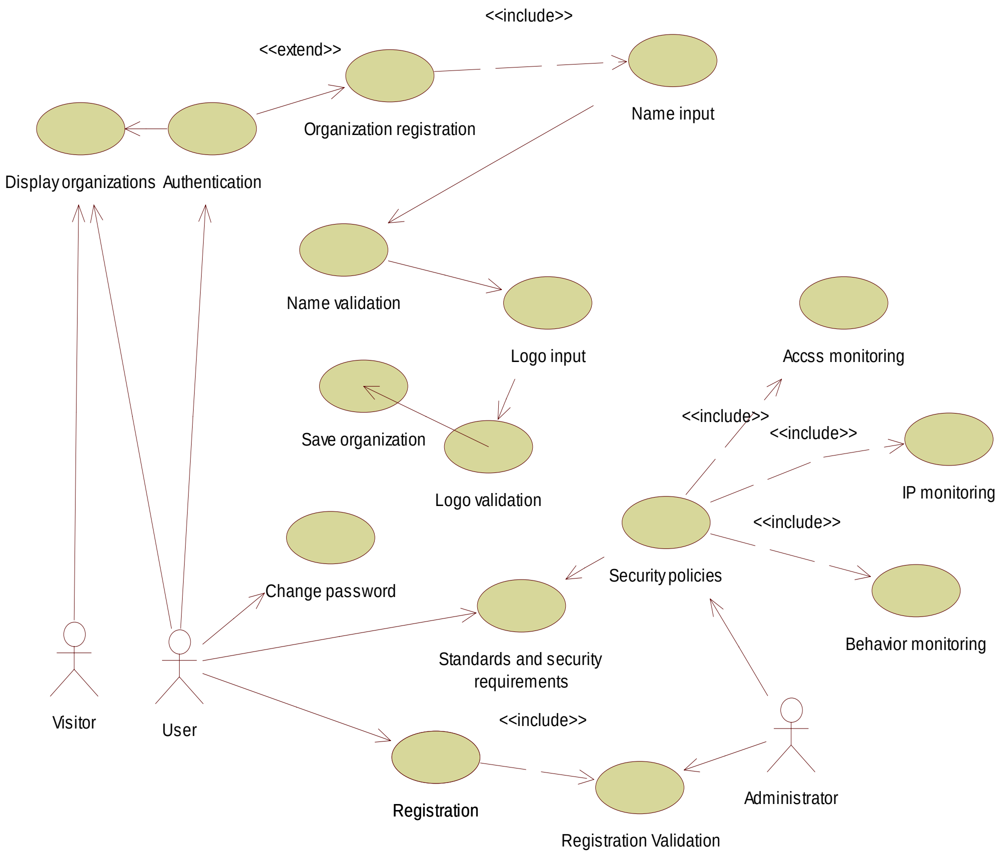

Figure 3. *AVIO use case*

From the use case shown in Figure 3 is determined the operations access matrix for AVIO application in Table 2.

Table 2

AVIO operations access matrix

Permit Role	Read	Identifiers orthogonality analysis	Add identifiers in the database	Validate users	Security policy management	Users management	Events management
Visitor	X						
Registered	X	X	X				
Administrator	X			X	X	X	
Application							X

The tables' access matrix that composes the database used by the AVIO application is presented in Table 3.

Table 3

Tables access matrix in AVIO application

Table User	App_FIRME	aspnet_Users	Log_EXCEPTII	Log_LIB	Log_Evenimente	Log_COMP	Log_ACTIUNI
Visitor	X						
Registered	X						
Administrator	X	X	X	X	X	X	X
Application	X	X	X	X	X	X	X

Guest user type has read access rights to the table organization identifiers stored in App_FIRME table without seeing which user entered them. In Table 4 are presented the permissions for the associated fields App_FIRME table for the Guest user type.

Table 4

Guest user access permissions to App_FIRME table fields

Field Operation	Name	Logo	ID	UserID
Read	X	X		

Registered type user has read access rights but also to introduce new organization identifiers if the analysis proves they are orthogonal with the existing identifiers from the database. Registered type user is restricted to delete only the identifiers self introduced and do not have permission to delete identifiers entered by other users.

Table 5 shows the permissions for the associated table fields App_FIRME table of the Registered user type.

Table 5

Registered user access permissions to App_FIRME table fields

Field Operation	Name	Logo	ID	UserID
Read	X	X		
Write	X	X		
Delete - restricted	X	X		

Administrator user has extended access rights to the App_FIRME table like registration of new orthogonal organization identifiers, reading, deleting the existing identifiers that are improper or modifying the existing ones. Modification involves the resumption of processing in order to determine the orthogonality to the changes carried on identifiers.

Table 6 shows the permissions for the associated table fields App_FIRME table of the Administrator user type.

Table 6

Administrator user access permissions to App_FIRME table fields

Field Operation	Name	Logo	ID	UserID
Read	X	X	X	X
Write	X	X		X
Delete	X	X		X
Modify	X	X		X

The Aspnet_Users, Log_EXCEPTII, Log_LIB, Log_Evenimente, Log_COMP, Log_ACTIUNI tables have reading rights for Administrator user and writting rights for the Application. Aspnet_Users table is used for users' management and the other tables form the AVIO application monitoring system having an informative role for the Administrator user type.

Access matrix

Access matrix implies the existence of the database structure because the access to the table's fields is done starting from the table's architecture that store data. The access type is imposed by the inclusion of access information in tables and by creating the auxiliary tables to achieve a high level of granularity in defining access rules. To achieve the access matrix is started from the permissions data to determine the access level to the table's fields from database. It is considered the database consisting of a set of tables T= $\{T_1, T_2, \dots, T_n\}$.

To each T_i table is assigned a set of records $I_i = \{I_{i1}, I_{i2}, \dots, I_{ik}\}$. The I_{ij} record is defined by a set of fields $C_{ij} = \{C_{ij1}, C_{ij2}, \dots, C_{ijm}\}$. There are access restrictions to the table level classified depending on the table's role in the software product ensemble.

Thus there are tables that are subject to reading operations and it allows:

- unrestricted access to read, this type of table contains non-confidential information aimed at informing users by offering for reading the stored data; the tables with unrestricted access to the data are used in information portals, government sites for instructing or other types of computer applications aimed at providing non-confidential information;
- partially restricted access to reading; this type of table contains information which are confidential, but also public information; such access is selected by the user's access level; if the user has access rights to confidential data the entire table is presented, otherwise only the public information is presented; these tables are used in applications that implement security systems, users and roles management for differential access to information and resources; the tables that have partially restricted reading access are used to record confidential details and the elements of public interest that are presented anonymously in the absence of access rights;
- reading restricted access, this table presents only confidential information that are presented to the users with access privileges; reading restricted access tables are used for recording confidential information; access is checked and data is encrypted in order to ensure a high level of confidentiality.

There are tables on which write operations are performed depending of the purpose of the software product, by allowing:

- unrestricted write access, records are made automatically by the software application or manually by the user;
- partially restricted write access means allowing the writing of only a set of table fields, the other ones receiving values automatically assigned through an algorithm; this type of table is used for sales where the items prices, VAT and the price without VAT are automatically filled; thus the user does not receive the write access for the entire table;
- restricted write access so only users with writing access perform this type of operation;
- partially restricted write access or restricted with validation; this principle is applied to partially restricted and restricted write access;

the users performing inserts in this type of table must have special access privileges as well as those that perform data validation.

There are tables on which modification operations are carried out allowing:

- unrestricted modification access in which the amendments are made by all registered users of the database or of the software application that uses the table;
- partly restricted modification access by preventing the amendment of high privileges access fields;
- restricted access to modify by preventing the amendment for all users necessary access rights to the table;
- partially restricted access and restricted access with validation to modify; this kind of access allows returning to the unchanged record if the change is erroneous or not accepted by the user that is responsible to validate the operation.

The delete operations must be performed by users with higher privileges in order to restrict the group of users who have access to this functionality.

There are tables that have mixed permissions access for reading, writing, modifying or deleting; such privileges presumes mixed access having a combination of access types for each presented operation. Table 7 presents an access matrix to the tables set T = {T_1, T_2, …, T_n} for the users set U = {U_1, U_2, …, U_p, …, U_t}.

Table 7

Access matrix to the tables set

Table User	T_1	T_2	…	T_{i-1}	T_i	T_{i+1}	…	T_{n-1}	T_n
U_1	X			X					X
U_2					X	X			
…									
U_{p-1}		X							
U_p		X			X			X	X
U_{p+1}	X	X		X	X	X		X	X
…									
U_{t-1}	X					X		X	
U_t		X			X				

There are several types of fields:

- which are used to identify an element from a collectivity after specific coordinates such as unique identification codes, personal identification number, registration number or other unique items;

- which is used to describe entities that contains C_{ij} fields necessary to identify the characteristics for the virtualized entity record I_{ij} containing fields type C_{ij}; the fields used for description are required to define the entity characteristics in order to determine the difference degree of the entities collectivity stored in the T_i table;
- it aims the state and dynamics of the C_{ijl} element belonging to the C_{ij} ¬ fields collectivity associated to the I_{ij} record; this field has a dynamic developed since it is subject to frequent changes that define the entity dynamic; this field type is used to identify the evolution of a currency pair on the market exchange, the roadmap management of an automobile or other activity types that require frequent changes;
- connecting entities from other tables that are designed to maintain the logical connections between entities; these fields are represented of unique identifiers that belong to other entities and are used to create logical connections between the two records;
- fields for the delimitation of the daily expenditure that are subject to change depending on the number of items purchased as a result of procurement transactions.

In order to apply the security restrictions for each field it is necessary to establish the conditions for editing fields. Such fields are:

- with unrestricted read permissions which are displayed to all users and serve the public interest in the database; the access to these fields is done with minimal privileges;
- with restricted read permissions to be displayed only to the users who have the necessary permissions to gain access to the data stored in respective fields; this field type contains confidential information or an increased importance level;
- with unrestricted write permissions that are written by all database users; these fields contain items of general interest and with a moderate importance level;
- with restricted writing permissions that are written only by users who have permissions to change that field; this field has high importance and in some cases contain confidential information;
- with unrestricted modify permissions that are modifiable by all the database users; these fields are used to store the dynamic data that changes frequently and have a moderate importance level;
- with restricted modify permissions that are modifiable only by certain database users that have access privileges to that field; these fields are used to store confidential data or with high importance level.

For the C_{ij} fields set associated to the T_i table is considered the basic operations matrix allowed for the U_p user shown in Table 8.

Table 8

Basic operations allowed by the U_p user

Field / Operation	C_{ij1}	C_{ij2}	…	C_{ijo-1}	C_{ijo}	C_{ijo+1}	…	C_{ijm-1}	C_{ijm}
Read	X	X		X				X	X
Restricted read					X	X			
Write		X							
Restricted write	X								
Modify		X			X			X	
Restricted modify	X			X		X			X
Work simulation		X		X				X	
Copy data	X			X					X
Export data		X			X			X	

In Figure 4 is presented a three-dimensional matrix that provides access control the T_i table fields.

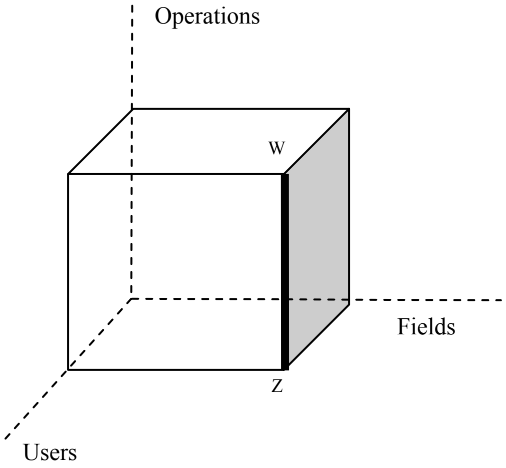

Figure 4. *Fields access three-dimensional matrix*

The WZ segment is represented by the operations set to be performed by all users. Figure 5 shows the three-dimensional arrays for the fields access to the T={T_1, T_2, …, T_n} tables. In order to optimize the access to the fields and to impose effective restrictions user roles are implemented in order to effectively create operations restrictions for users. Access permissions to the table fields are saved in the database by the administrator and associated to the user roles. By implementing user roles the permission tables are tighten thus requiring a number of permissions set equal to the number of roles from the database.

The three-dimensional role-based access matrix is smaller and more manageable. Addressing the access control through roles is effective when the permissions allocation for each user is redundant, having multiple users with the same sets of allowed operations.

For proper management of fields access it is necessary to use triggers tables or using the programming language that implements the database.

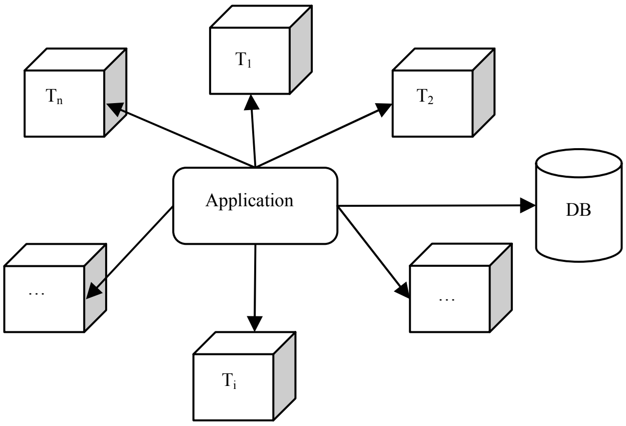

Figure 5. *The permissions set for the T tables collectivity*

The permissions set for the T tables collectivity is representative for the access to information stored in the database by the fact that is implemented an access level to the field level that allow an effective control of the operations on the database resources.

The three-dimensional access matrix provides the possibility to identify all user actions and their control if necessary. Its implementation is important in the context of maintaining a uniform way to control the access to the resources.

Users classification

Online applications, through their complexity, offer a large variety of interactions types which correspond to the users types. The users consult the online applications to get information. There are online applications that are used for making complex transactions and confidential nature such as:

- document management through storage and loading of the documents on a computer server; this application is useful because it provides high availability of the documents regardless of the used terminal; the applications that provide this service have the option to view online documents, edit them, grammar checking or downloading or saving the modified document on the working terminal; there are security restrictions implemented to maintain the confidentiality of the stored documents; is also implemented the functionality to change the confidentiality of documents into a public one;

- online budget management through introduction of monthly expenses and income; this type of online application is useful to clearly identify and eliminate unnecessary spending; personal data regarding income and expenses must be protected by an effective security system because the database privacy loss lead to the disclosure of a large volume of highly confidential information;
- managing bank accounts through e-banking applications; such applications provide the user with a large number of banking transactions such as management of deposits, payments to vendors, making bank transfers or view statements; to preserve the confidentiality is necessary to effectively manage the system security and using of information encryption mechanisms and effective authentication mechanisms that take account of the latest types of threats;
- virtual stores that offer the products viewing functionality manage shopping cart and online payment; this kind of application is useful for making online purchases of various products, management of orders and views the shopping history.

Users types in complex applications that have a high level of confidentiality is set by the access level they need to operate. There are users who:

- visualize the presented information and use them in their activities;
- add information in the database; these users need differentiated permissions to determine in which tables have the permission to add data and what fields are affected by the added records;
- change information in the database; these changes occur due to inventory differences or changes made in documents or other factors of influence; in order to make the changes is needed to establish the tables and fields modifiable by that user; this is achieved by setting the necessary permissions for the user to efficiently carry out its proposed activities; changes made are at the field or table level and, in some cases, involves changes in other fields or tables as side effects;
- delete the information from the database that are no longer useful or have been incorrectly entered; this type of user needs high access privileges and well defined to select tables and fields that has access

to; by implementing the deletion logic mechanisms are created the premises of a better database management and the risk of permanent loss of the information are reduced;

- validate changes; this type of user has a high access level with the task of validating the changes, additions or removal operations initiated by other users; the user role is necessary in the context in which the application is working with valuable information such as bank details or scholarship indicators to ensure the integrity and accuracy of the database; the privileges are given depending on the position and the competence of the user having areas where it has the needed expertise to validate the fields or table changes;
- manage the application and have the highest access level of all users; this user type has administrative rights on the database bringing structural changes and managing data stored in the database, has user management rights and can activate and deactivate user accounts.

Figure 6 shows how to manage changes in the database done by different user types; U_1user has Administrator rights, U_2 have validation rights and U_3, U_4, U_5 and U_6 users have modification rights. Figure 6 shows a scenario limited to six users to differentiate the access levels and user roles.

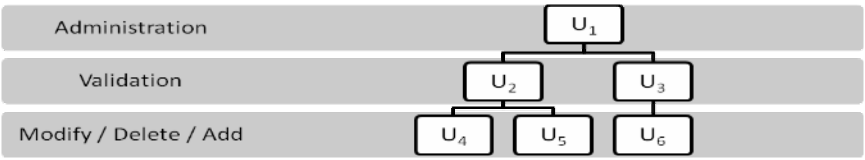

Figure 6. *Operations validation performed by users*

The types of users are ranked according to the privileges types that they receive in the distributed application. In Figure 7 the users' hierarchy is carried out according to the group permissions.

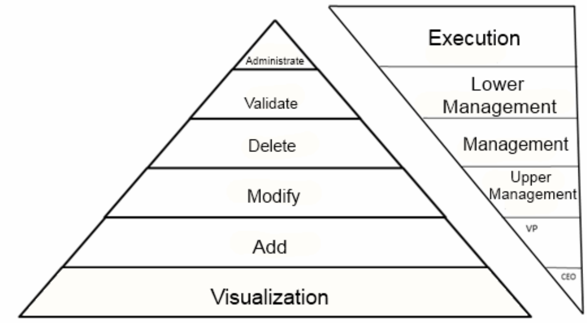

Figure 7. *User type classification in the organization hierarchy*

The user types are determined through direct connection with the users' roles and their relationship with the database entities. The tables and fields access restrictions are managed according to user type. If the user does not have the privileges to visualize, change or delete a field or a table entry the operation is suspended and the action is recorded in the logging table.

Conclusions

Online applications have the advantage of providing services of increased complexity in a distributed environment. Online applications efficiently work with databases, make use of services and have intuitive user interface. There are online applications that present general interest information, but also online application that process confidential information requiring access restrictions.

The data access issue is important to be treated in order to mark the boundaries between strictly confidential and non-confidential data. This is done by analyzing the information flows from the online application. It is necessary to establish access methods to the database tables and table fields.

By using the data access matrix it is provided a clear understanding of the access privileges used in the online application. It is necessary to establish access permissions for each user type to correlate the permissions with the information which provides access to. Data access matrix is a stable indicator of the user access level presenting small variations to a small number of operations available for a small number of users, but high variations in a large number of operations available for a large number of users.

The using of the three-dimensional matrix representation as a method of access privileges is significant in the context of assessing the number of users and permissions.

AVIO application is made to examine the organization identifiers orthogonality. The processes implemented for determining the difference is only accessible to registered users. The application offers the possibility to view the identifiers stored in the database. The access setup process is based on the information confidentiality. For this purpose only the administrator has access to the running information and the registered users of the AVIO application.

The costs for data access matrix are low because it involves an analysis of the flows of the online application through identifying the operations in the use cases. The cost of access privileges identification appropriate to user types is lower using the access matrix in the phase of the specifications development than in the implementation phase of the system security.

Acknowledgements

This article is a result of the project „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Programme for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.

References

Roşca, I.Gh., Ghilic-Micu, B., Stoica, M. (2006). *Informatica: societatea informaţională: e-serviciile*, Editura Economică, Bucureşti, ISBN (10) 973-709-266-X

Cotfas, L., „A Genetic Algorithm and GIS based solution for public transport networks", *The Proceedings of the Ninth International Conference on Informatics in Economy*, 2009a, ISBN 978-606-505-178-2

Cotfas, L.A., „Advanced personalization of location based services", *4th International Conference on Knowledge Management: Projects, Systems and Technologies*, pp. 31-34, Bucharest: "Carol I" National Defence University, 2009b, ISBN 978-973-663-783-4

Vintilă, B., Pavel, S., „Assisted Design, Development And Evaluation Of Citizen Oriented Collaborative Applications", *Journal of Applied Collaborative Systems*, vol. 2, no. 3, 2010, ISSN 2066-7450

Dicţionarul explicativ al limbii române (1998), Academia Română, Institutul de Lingvistică „Iorgu Iordan", Editura Univers Enciclopedic

Palaghita, D., „Quality characteristics of open source components", *Open Source Journal*, www.opensourcejournal.ro, 2009

Ivan, I., Boja, C. (2004). *Metode statistice în analiza software,* ISBN 973-594-498-7, Editura ASE, Bucureşti

Doinea, M., Ciurea, C., Dumitrache Marilena, „Collaborative Environmental Security Facing the Challenges of the Economic Process Development", *Proceedings of 17th International Economic Conference – IECS 2010*, „The Economic World Destiny: Crisis and Globalization?", May 13-14, 2010, Sibiu, Romania, Special Issue of Revista Economica, Lucian Blaga University of Sibiu

Doinea, M., Pavel, S., „Security Optimization for Distributed Applications Oriented On Very Large Data Sets", *Informatica Economica Journal*, Vol. 14, No. 2, 2010, ISSN 1453-1305

Cotfas, L.A., Diosteanu, Andreea, Smeureanu, I., „Knowledge Dynamics in Semantic Web Service Composition for Supply Chain Management", *Journal of Applied Quantitative Methods*, Vol. 5, No. 1, 2010, ISSN 1842-4562

Ivan, I., Vintila, B., Ciurea, C., Doinea, M., „The Modern Development Cycle of Citizen Oriented Applications", *Studies in Informatics and Control*, Vol. 18, No. 3, 2009, ISSN 1220-1766

Ivan, I., Vintila, B., Ciurea, C., Doinea, M., „Citizen Oriented Informatics Applications development Cycle", *Ekonomika, Statistika, Informatica, MESI*, No. 4, 2009b, ISSN 1994-7844