International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Concept of Graphical Interface of a Medical Information System (CheBHosp)

Mohamad Ali Cheaito[1], Marwan Cheaito[2]

[1]Lebanese University, Faculty of science I, Computer Science Department, Hariri University Campus, Beirut-Hadath Lebanon,
[2]Lebanese University, Faculty of science V, Computer Science Department, Nabatieh Lebanon

Abstract: ***The objective of our study is to review the basic concepts and the implementations of an interface of a health information system [1] for handling information within a hospital. The realization of the graphical interface is one of the points requiring most of the time in the development of a computer application. To make the use of the application easy, pleasant, and adaptable to the user's role, we have decided to use in the graphical interface event-based objects, called buttons. When buttons are clicked, they allow the access to different forms. These visual elements make the interface more intuitive. Moreover, the employees of a hospital are usually far from recognizing easily the use of computer software and tools. The training formation of an employee requires the use of computer software and tools for medical treatment. That's why; the use of the graphical interface that considers the constraints between forms and buttons becomes more intuitive and easy. On the other hand, the choice of the graphical interface should be adequate with the structural organization of the hospital, in order to share the interface between two major processes: handling data (inputs) and requests presented as reports (outputs) [2].***

Keywords: Graphical Interface**,** Hospital Information System, Patient, Data Base

1. Introduction

Our work will be in general on the border line between informatics and medicine. This arises the question of using a graphical interface for handling information circulating within a hospital. The medical information system is a system containing many subsystems like accounting system[3], [4], storage system, human resources system, administrative system… that communicate with each other by using patient identity [5]. For the graphical interface of management of information, it is useful to share information in distributed databases depending on its use [1], [6].The graphical interfaces must consider the role of user authorization affecting the subsystem included in the global system [7].The graphical interface should respect the constraints and the criteria of a language interface and a data definition language. A relational database management system such as SQL [8] must ensure a proposal of data and types allowing easy manipulation of graphical interfaces and users permissions.

2. Partition of Graphical Interfaces

We divide the data into four sub-systems: patient management system, accounting system, storage system, and administrative system. In each sub-system, the set of entities (things in the modeled world) which are the base of information of the sub-system such as laboratories information, pharmacological information ...should be integrated together by a system whose management becomes easier when we use the administrative system especially in assigning user's roles and describing their work (a user who works in the laboratory service has a different role than a user working in the medical imaging service).The separation between the input interface and the modification interface is a direct link for the description of the user's work role. Sometimes a user may have an entry role to input data without having the right to modify data affecting the liability of a more precise medical procedure (figure 1) [9].

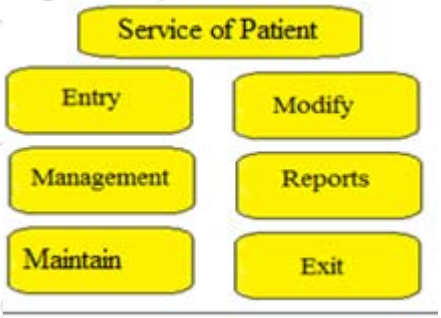


Figure 1: Graphical interface of patient service

The historical access control models have led to the proposal of access control roles (Role Based Access Control – RBAC) whose principle is to introduce an indirect level between users and permissions [10].

The graphical interface is generally intended for users often non-computer specialists. This interface must be adaptable to non-computer users. The button is a simple mean for manipulating computer tools which allow non-computer users to easily complete their tasks with a little guidance.

3. Graphical Interface to Define the Role of a User

Several access control models organize users' rights depending on the business processes in which they intervene. The partition of tasks of the same process between several actors or users ensures the principle of separation of duties [11]. This principle is generally introduced in the structured models through the notion of "constraint" which limits the permissions granted to users. The literature distinguishes two broad categories of constraints:

- The dynamic constraints are evaluated according to access requests, through the execution on the screen and the opening of sessions [12].

- The static constraints are evaluated once for all and do not depend upon the execution on the screen.

A constraint can generally belong to either of the two categories: the dynamic category, if the constraint covers topics representing the activities of the users on the system, or the static category, when the constraint involves the users themselves that are passive entities from the access control point of view.

As it is difficult to bring together into a single model that is coherent and homogeneous the diverse needs for organizations security, the authors nowadays offer logical frameworks (figure 2) that formalize and develop access control models [13]. This formal approach is much less pragmatic than that motivated in first by the use and administration of access control. A multiple logical frameworks are proposed to model access control [14], [15], [16], [17], [18], [19], [4].

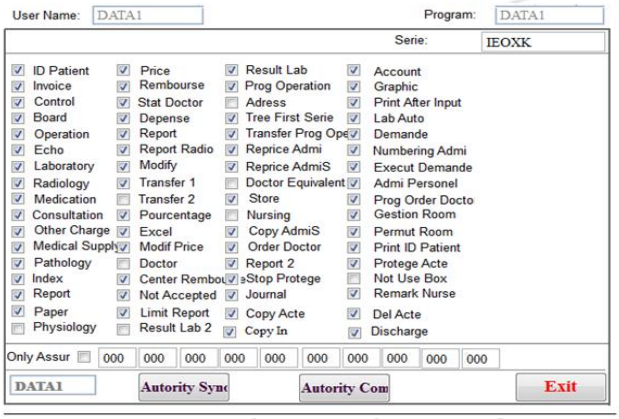


Figure 2: Gaphical interface to assign roles for a user in a medical system

4. Graphical Interface for Filtering Reports

The programmer must ensure for the customer a set of reports that respect the needs of his/her company, hence the necessity of making reports for one of the users and not the others. Each report is identified by a natural number which is assigned to a user by a "checkbox" object preserving the checked value into a database (figure 3)[20], [21]

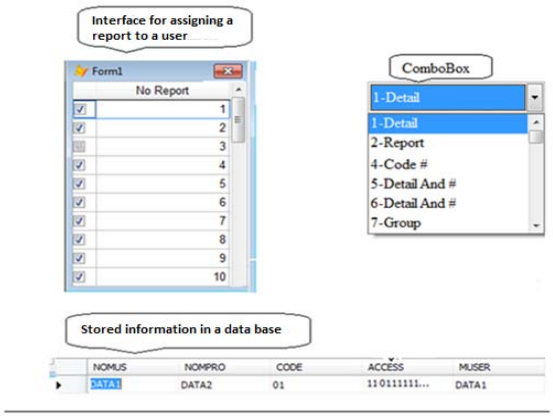
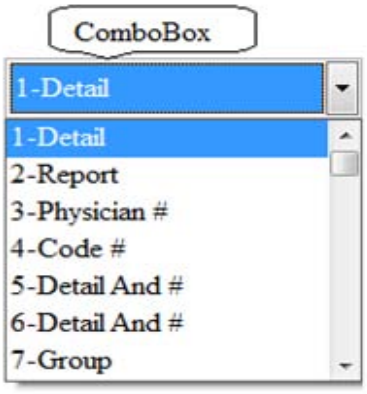


Figure 3: Making a report for a user

The natural number "i" of the i^{th} report corresponds to the i^{th} character of the access field 1indicates the validation of checkbox and the access field 0 indicates the non-validation of checkbox. To realize this, the choice should be assured by a "comboBox" object that is filtered according to the chain of characters stored in the access field while for other users the "comboBox" which occurs should be as follows:



Functions of comboBox manipulations:
Function for making a report for one user:

```
Procedure SecurityReport(Comb1:ComboBox,
MaRCode:String)
Variables N:Integer, N1:Integer, I:Integer, MaReq:String,
MyTab:DataTable, Tab(N):String
 N ← Comb1.Items.Count
 N1 ← N
 For I from 0 To N
TRep(I) ← I
EndFor
MaReq← "SELECT TOP 1 ACCESS FROM
[STAFF].[dbo].[TACCESS] " + _
"WHERE [NAMEUS] = „"+NameUser+""" + _
" AND [NAMEPRO] = „"+NameProg+""" + _
" AND [CODE] = „" +MaRCode+"""
callLoadTable(MyTab, MaReq)
 If (MyTab.Rows.Count> 0) Then
 For I from 0 To N - 1
Tab(I + 1) ← Comb1.Items.Item(I)
EndFor
 N1 ← 0
 For I from 1 To N
If (Mid(MyTab.Rows(I).Item("ACCESS"), I, 1) = "1") Then
 N1 ← N1 + 1
TRep(N1) ← I
 End If
EndFor
callComb1.Items.Clear()
For I from 1 To N1
callComb1.Items.Add(Tab(TRep(I)))
EndFor
 End If
 End Procedure
```

Function for loading data from a data base relying on access request:

```
ProcedureLoadTable(MyTab:DataTable, MaReq:String)
VariablesMyCQ:NewSqlCommand(MaReq, ConSQL),
J:Integer, ex:Exception
MyCQ.CommandType←CommandType.Text
callMyCQ.Connection.Open()
callMyTab.Load(MyCQ.ExecuteReader(CommandBehavior
.CloseConnection))
```

```
 For J from 0 ToMyTab.Columns.Count - 1
MyTab.Columns(J).ReadOnly← False
EndFor
 Catch Exception ex
Write("Request Error:" + Chr(13) + MaReq)
 EndCatchException
 EndProcedure
```

5. Graphical Interface for Data Manipulation

The graphical interface is a mean that easily allow the manipulation of stored data in a database management system [22], [23] (Figure 5).

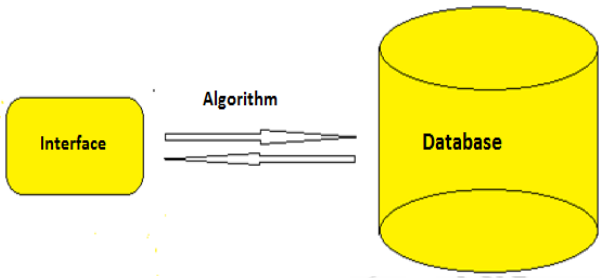


Figure 4: Schema between the graphical interface and the stored data

Method for selecting a patient:

To select a patient that is stored in a table called "Patient" within a database, we must define an object named "SearchAdmi" containing the following components (figure 5):

- a form
- fourtext objects known as TextBox
- a grid

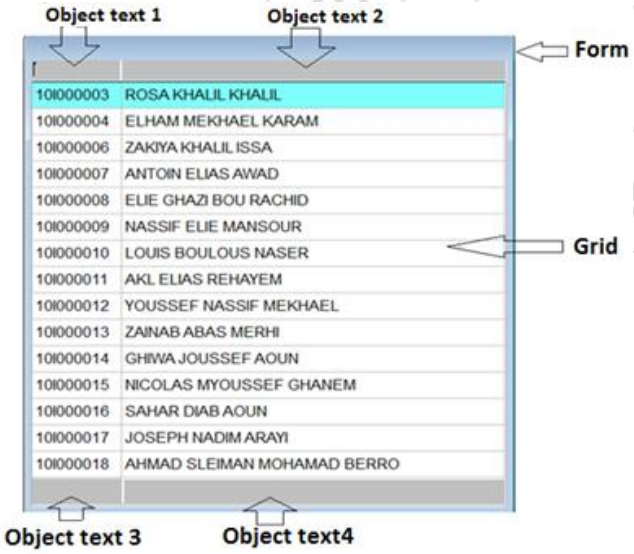


Figure 5: Interface for selecting a patient

The object "Search Admi" should assure a selection of a well determined patient. For example in figure 5, when the user enters the character '1' in the object "text1" the grid displays the first 15 patients that their identity begins with character '1' and if then the user enters character '0' the grid displays the first 15 patients that their identity begins with '10' and so on. The same process applies for entering data in object "text 2". On the contrary, the object "text3" and the object "text4" allow filtering of the set of patients having in their text at any position the set of entered characters in the object "text 3" or the object "text 4",

SQL query that corresponds to object "text1":
“SELECT TOP 15 ADMI, NAME FROM [DATA1].[DBO].[PATIENT] "+
" WHERE ADMI Like “+Text1+”+‘%’”
" ORDER BY ADMI"
SQL query that corresponds to object "text4":
“SELECT TOP 15 ADMI, NAME FROM [DATA1].[DBO].[PATIENT] "+
" WHERE NAME Like „%“+“+Text1+”+‘%’”
" ORDER BY NAME"

This object provides to the user a manipulation of data not only based on coded information but also on more detailed information. Moreover, it allows internal refreshment of information between the user and the database.

6. Conclusion

The domain of health care is nowadays seen as a domain that should be computerized with a large scale. The complexity of medical information is an obvious brake. It seems clear that the current health information systems must evolve in order to incorporate a flexible graphical interface to be used by non-special computer users. Encodings incorporate ambiguities on the choices carried out by the user and derives from the heavy training in the company in question. For this reason, our proposition is based on the manipulation of data by making a simple connection with the database server and allowing the refreshment of data in a simple way. The user roles depend on the specific case of a hospital center that provides an update on the uses of an information system.

References

[1] P. Degoulet, M. Fieschi. Informatique médicale. Elsevier Masson, 1998. : 1998vb
[2] E. K. Fromme, T. Kenworthy-Heinige, M. Hribar, Developing an easy-to-use tablet computer application for assessing patient-reported outcomes in patients with cancer. Supportive Cae in Cancer June 2011, Volume 19, Issue 6, PP 815-822.
[3] S. Langer, J. Wang, User and system interface issues in the purchase of imaging and information systems, Journal of Digital Imaging August 1996, Volume 9, Issue 3, pp 113-118.
[4] C.Lovis, S.Spahni, N. Cassoni-Schoellhammer, A.Geissbuhler. Comprehensive management of the access to a component-based healthcare information system. Studies in health technology and informatics, vol. 124, pages 251_256, 2006. :2006tn
[5] C.O.Bagayoko, J.-C. Dufour, P. Avillach, C. Quantin, M. Fieschi .Reflections on the patient identification in the Healthcare Information Systems, ELSEVIER MASSON, IRBM 29 (2008) 302-309.

[6] M. A. Cheaito, M. Cheaito, The Concept of a Medical Information System By Rate of Use (CHEBHosp), Health Informatics-An International Journal (HIIJ), Vol 4, No 1/2, May 2015, Page:23-33, DOI:10.5121/hiij.2015.4201
[7] M. Fieschi, Les données du patient partagées : la culture du partage et de la qualité des informations pour améliorer la qualité des soins. Rapport au ministre de la santé. 2003. Disponible sur le site : http://www.d-mp.org/docs/Rapport fieschi.pdf, consulté le 20 octobre 2007.
[8] Weller, Bart, Beginning SQL Server Modeling 2010 ISBN 978-1-4302-2752-6 eBook.
[9] M. Seidl, M. Scholz, C. Huemer, G. Kappel, UML @ Classroom An Introduction to Object-Oriented Modeling ISBN 978-3-319-12741-5, 2014 eBook.
[10] D. F. Ferraiolo, R. D. Kuhn, R. Chandramouli, Role-based access control. Artech House Publishers, 2003. ii, 24, 25, 27, 28, 29, 34, 35, 39, 40, 51, 56, 70, 71, 77, 105, 117, 130, 137, 138, 139, 154, 162, 204, 241.
[11] M. J. Nash, K. R. Poland, Some Conundrums Concerning Separation of Duty. In IEEE Symposium on Security and Privacy, pages 201–209, 1990. 32, 38, 39, 45.
[12] A. Klein, A.Shaban-Nejad, G. W. Rose, A. J. Forster, D. L.Buckeridge, C. J. Baker, Semantic quering of relational data for clinical intelligence: a semantic-based approach Journal of Biomedical Semantics March 2013, 4-9
[13] S. Harkins, Susan, Reid, Martin, Access to SQL Server 2002 ISBN 978-1-4302-1573-8. 2002 eBook.
[14] J. Y. Halpern, V.Weissman. Using First-Order Logic to Reason about Policies. In CSFW'03: 16th IEEE Computer Security Foundations Workshop, Pacific Grove, California, pages 187–201. IEEE Computer Society, 2003. 46, 47, 51, 63, 114, 128, 146, 201
[15] E.Bertino, B. Catania, E. Ferrari, P.Perlasca, A logical framework for reasoning about access control models. ACM Transactions on Information & System Security, vol. 6, no. 1, pages 71–127, ACM Press, New York, 2003. 46, 47, 51, 52, 63, 147
[16] S. Jajodia, P.Samarati, M. L.Sapino, V. S. Subrahmanian, Flexible support for multiple access control policies. ACM Transaction on Database Systems, vol. 26, no. 2, pages 214–260, ACM Press, New York, 2001. 46, 47, 52, 62, 63
[17] M.Abadi, . Access Control in a Core Calculus of Dependency. Electronic Notes in Theoretical Computer Science, vol. 172, pages 5–31, Elsevier, Amsterdam, Avril 2007. 46, 47, 50, 62, 63
[18] J.nDeTreville. Binder, a Logic-Based Security Language. In SP'02: IEEE Symposium on Security and Privacy, page 105, Washington, 2002. IEEE Computer Society. 46, 47, 50, 63
[19] M. A. Al-Kahtani, R. S. Sandhu. Rule-Based RBAC with Negative Authorization. In ACSAC'04: 20th Annual Computer Security Applications Conference, pages 405–415, Washington, 2004. IEEE Computer Society. 46, 47
[20] Wright, Heather, Begininng Visual Basic 2005 Express Edition ISBN 978-1-59059-622-7 2006 eBook.
[21] Thomsen, Carsten, Database Programming With Visual Basic .NET 2001, ISBN 978-1-4302-1132-7, 2001 eBook.
[22] E. LeopoldoBertossi, Consistent query answering in databases. SIGMOD Record, vol. 35, no. 2, pages 68–76, ACM Press, New York, 2006. iv, 87, 206
[23] R. Fagin. Inverting schema mappings. In Stijn Vansummeren, editeur, PODS'06: 25th ACM SIGACT-SIGMOD-SIGART Symposiumon Principles of Database Systems, Chicago, Illinois, pages 50–59. ACM Press, 2006. iv, 85, 87, 170, 206, 207