Interest Square Unit As
A Visual Basic Application
Generator

By

Sameh Ahmed

Sereen Hatem

Special Topics in Computer Science

In Object Oriented Design

For Senior project

(Individual Studies)

Dr. Raiek ALNAKARY

Tuesday 22nd of December, 1998

Dedication:

We would like to dedicate this paper and project for all the students who want to study Object Oriented Design and Programming. We also, would like to dedicate it to our parents who did their best in supporting us and making all our needs available and handy. We would like to dedicate this project for anyone who is interested in studying a new method in software programming. Finally, we would like to dedicate this project for the ones that love us and support us morally and spiritually.

Introduction:

First, we would like to thank ***Dr. ALNAKARI*** for the effort that he have exerted in teaching us how to program in Object Oriented fashion and the fact that he provided us with a state-of-the art latest technology software applications such as **OMTOOL** and **RATIONAL ROSE.**

This paper and project's main purpose is to divide and define the Microsoft Visual Basic controls according to the ***HAYAWIC UNILOGIC*** and ***INTEREST SQUARE UNIT*** which are created and invented by ***Dr. ALNAKARI***.

Outline:

I. Hayawic Logic and ISU and the tools to represent them:

A. What is Hayawic Unilogic ?

B. What is Interest Square Unit?

C. What are the tools used to represent the Hayawic Logic?

II. The Microsoft Visual Basic and the Hayawic Logic Theorem:

1st. The History of the Microsoft Visual Basic.

2nd. The application of ISU division to the Microsoft Visual Basic controls.

Preface:

To us, studying the theorem of the ***HAYAWIC UNILOGIC*** that ***Dr. ALNAKARI*** have introduced to us was not a hard job to do, because it puts everything in the usual ***form*** that we know. Cause it refers everything to the word ***form*** which can describe all the components of this vast universe.

The hard part of the course was to implement the application that carries out the functions and the features of both the ***HAYAWIC UNILOGIC*** and ***INTEREST SQUARE UNIT*** methodology. Moreover, getting documents to support our aims. Therefore we have decided to reach the nearest public internet café (as they are know here in Egypt) in order to locate suitable documents that could help us.

The second part, was to get used to the tools that Dr. ALNAKARI have introduced like ***OMTOOL*** and ***RATIONAL ROSE***, which were new as useful to our project and education, in addition to our own knowledge.

We all know that everything in this universe can never be the same as another object due to several reasons. Though some objects may seem similar to some other objects, but in their structure they may differ, and if they are similar in structure, they would differ in the looks. This is very much apparent in twins, for example, where you could never find identical twins in this universe. There would be differences, that are not usually noticed from the first time, either in body features or in behavior. Therefore, we need to find something through which we can identify all the universes' components and none-components. This 'something' itself should be described inclusively within this 'something'. Well, brainstorming and going through encyclopedias and books did not help much finding what this 'something' might be. But then we have discovered that there is a theorem that is known as "***Hayawic Form Unilogic System***" that was founded and developed by Dr. Raiek Al-Noukari from the Damascus School of Logic.

The theorem states, simply, that we can identify all the universe's components as a matter of "***form***" cause everything in this universe has a form that is different than anything else. Meaning that all the things we feel, see, eat, drink,…etc., have a particular form that is different than any other form. The ***Hayawic Theorem*** extends the division of from to:

1. **Dynamic Form**: a form that changes by the time; like human beings, animals, and food

2. **Static Form:** which is the opposite of the first one; it does not Change due to any reason.

The ***Hayawic Theorem***, makes the use of interest square that is divided into four sub-squares. This square is called ***Interest Square Unit*** the sub-squares are:

- **The Isolated square**: contains all the negative items that we are, or would deal with. Moreover, these items are usually isolated in manner
- **The Conflict Square:** contains two contradictory items, that may, or may not, cancel each other.
- **The Coexisting Square:** here, we have two different items, but they Coexist in life mutually.
- **The Unifying Square:** is where all the items exist together in harmony.

Isolated	**Unifying**
Conflicting	**Coexisting**

Figure 1: The Interest Square Unit

The Interest Square Unit is mainly a measurement unit which can be used to obtain quantity and quality measurement for any interest, and according to the division presented in figure 1 it is considered as a means of organizing.

The Movement of any element from one square to another depends mainly on the interest of the item. For example, a child, when he/she is young, usually rejects the idea of sharing his/her toys with the rest of the children; therefore he is ***isolated***. But when he/she starts to grow up, they start sharing certain toys with the rest of the children; this indicates that the child became entangled with the surroundings, but not the intended level cause now the child is in the ***conflicting*** state. When reaching teenage the young man/woman would live in harmony with his/her mates

each on his/her own way of thinking which indicates that the young man/woman are now in the ***coexisting*** status. Finally, when this teenager grows and become mature enough, he/she would learn that life is much more simpler than what he/she used to think in the past and by that he/she reaches the ***unifying*** status.

We have used the ISU model and applied to the Microsoft Visual Basic™ components and to generate an application that wold make use o this division and try, as much as possible, to generate, with the use of other methods, a visual basic application that is described by the user him/herself. In order to do that, we have returned to he basic components of any visual basic applications and classified these components according to the ISU model. This division will make programming applications into **Object Oriented** methodology an easier job for almost all computer users who might, and might not, be programmers nor computer users.

One of the methods that are used to represent the **Interest Square Unit (ISU)** module is the **Unified Modeling Language (UML)**. **Unified Modeling Language** is a third-generation method for specifying, visualizing, and documenting the artifacts of an object-oriented system under development. represents the unification of the Booch, Objectory,

and OMT methods and is their direct and upwardly compatible successor. UML was founded for the following reasons:

1. These methods were already evolving toward each other independently. It made sense to continue that evolution together rather than apart, thus eliminating the potential for any unnecessary and gratuitous differences that would further confuse users.

2. By unifying these methods now, we could bring some stability to the object-oriented marketplace, allowing projects to settle on one mature method and letting tool builders focus on delivering more useful features.

3. We expected that our collaboration would yield improvements in all three earlier methods, helping us to capture lessons learned and to address problems that none of our methods currently handled well.

In this project we are trying to use the **ISU** theorem to generate visual basic applications that depend mainly on user typed text. The program will identify all the English computer nouns and verbs that are typed in by the user, which are stored in a dictionary, then the computer will create the objects that are related to the computer nouns and the operations representing the computer verbs and place them in a visual basic application. Because creating such an application is creating a **lexical analyzer** that would compile all the **keywords** and its **parameters** into a working fully functional application, we have

restricted, for the mean time, our work to four simple arithmetical functions which are namely: **addition, subtraction, multiplication, and division**. On the other hand, we have made it possible for the user to type any sentence that should contain one of the pervious words that would refer to arithmetical operations. The program will, after typing in the user's sentence, show the distribution of the main components of the expected user's sentence according to the ISU model. The following step is presenting the user to a form that would request the user to specify the main mathematical operation that the user is interested in. The final output is a form that have input text boxes for two different operands, and a resultant operand text box.