Entity Complexity of Informatics Application Interface Type

Ion IVAN
Marius POPA
Economic Informatics Department
Academy of Economic Studies Bucharest, Romania
Saha PRIYATOSH
PhD student, Bangladesh

Abstract: It is defined the informatics application interface concept and it is presented the role of this one in software using. There are pointed out the interface characteristics and the factors which lead to their efficiency. The software interfaces are classified on the base of many criteria. There are defined evaluation indicators for interface complexity that build the picture regarding operation facility of informatics application and their adaptation to new problem requirements.
Keywords: complexity, interface, entity.

1. Informatics Application Interfaces

Informatics application interface represents the means through whose the user operates the resources of a computer. From the structural point of view, the software interface is formed by text, graphic symbols and multimedia elements.
Through interface there are made the following operations:

- Data gathering introduced by user in order to be processed;
- Information displaying in order to be analyzed by user in different forms: text, charts, sound, animations;
- Action transmission to the computing resources trigged by user.

Software and hardware generation evolution lead to many stages of software interfaces development. For classic informatics applications, the interface structure is based on defined requirements. For modern informatics applications, the complexity of implemented requirements and used technologies caused the radical modification of interface structure and functions.

Having in mind the complexity increasing of the problems that must be resolved, technology and hardware evolution, it is necessary re-structure the design activity of software interfaces such as the maximum level of efficiency will be got.

The interface efficiency is determined through evaluation metric system for interface quality characteristics. Also, interface testing and validation for users with heterogeneous informatics background lead to conclusions regarding the interface efficiency.

Complexity increasing of the problems resolved with computer determines the complexity increasing of the interfaces used in informatics application developed for this end. The interface efficiency increasing supposes simple use of this one. The key factor in order to reach this objective is repetitive operation number reducing.

Another increasing factor of the interface efficiency is feedback supplying by the system when an user makes an action. The feedback has the error message form or admonition sound.

The action sequences must be organized in groups structured in parts: start, middle and end. The user must have the satisfaction of operation ending trigged by him and to get the traced result. Also, the users must control the system. Thus, the system has to respond to the actions trigged by user.

2. Interfaces classification

There are many criteria for interface classification.

For homogenous criteria, it builds:

- interfaces targeted on homogenous groups – the objective is the same for all the members of the collectivity; through concepts, principles, techniques and methods specific to target group are presented;
- interfaces that are addressed to heterogeneous groups – the student collectivity from an university is characterized by the following elements:
  - high level comprehension degree;
  - large problem diversity.

Within collectivity, there are identified student groups depending on interest fields: social, human and technique sciences, medicine etc. The interface of an informatics application for student's assistance in order to get knowledge comprises heterogeneous information from the point of view of content.

Depending on criteria, the following interface types are identified:

- interfaces that consider the color combinations, picture dimensions, position of theses ones on display – the interfaces of this type trace user's attention catching to the aspect with big importance that must be considered in application using;
- interfaces that consider only functional aspects – the informatics application that have this characteristic are extremely specialized in operation execution.

According with used means criteria, the following interface types are distinguished:

- text interfaces – suppose character using exclusively; it is necessary that the user to have knowledge regarding the standard character combinations used in application management; Also, a help tool is offered to the user where standard character sequences are found;
- graphic interface – represent an evolution of text based interfaces and they appeared as a result of advanced operating system development, with graphic facilities and available to very large user categories; this kind of interfaces suppose graphic symbols use associated with color tones in order to catch the user's attention; also, the building way of the interfaces is standardized from the point of view of functionality;
- mixed interfaces – they are combinations between text interfaces and graphic interfaces; a mixed interface contain graphic elements that include text; there are many types of graphic elements, called controls in programming activity: edit fields, text fields, check and radio buttons lists etc;
- multimedia interfaces – represents the actual stage in interface evolution; they are mixed interfaces to whom animated graphic elements, sound and video records were added.

From the point of view of content, the interfaces are:

- static – are characterized by a content which is not changeable in time, the validity and opportunity being a long term one; this kind of interface needs a small effort to update the content;
- dynamic – are interfaces with a content modifiable in time through:
  - opportunity decreasing of the represented information;
  - interactive character of the interface: user's possibilities to request, add and modify data.

If the used equipments are considered, then there are identified interfaces for:

- desktop computers – they are classical computing systems; the peripheral output equipments have the biggest representation surface of interfaces, an important thing for complex applications;
- mobile phone – it represents a voice communication device through communication standard implementation: GSM, CDMA etc;
- PDA (Personal Digital Assistant) equipments – they represent mobile devices with similar functions with a computer, having a small processing power;
- portable computers – they are computing systems with a small dimensions and weight; they have the mobility characteristic; they transform into desktop computer when they are connected to electrical network and to Internet through cable connection.

Depending of continuity criteria, there are:

- various interfaces – new interface design and implementation don’t take into consideration the old versions; the new interface has a new configuration;
- continuous interfaces – new interface design and implementation are made on the base of old interface; intuitively, the user identifies the functional elements relying on what he knew about old version.

Depending on interaction with user, the interfaces are:

- command language – the user formulates a request, introducing standards strings called commands;
- menu – the user has the possibility to choose from more variants the version which assures to him the objective achievement;
- with pre-establish format – the user is forced to introduce the data requested by the system in asked form;
- wizard assistants – they are applications that help user to accomplish some operations; the wizard assistant is an interface succession in which the user choose and introduce data.

Depending on complexity, the interfaces are classified in:

- complex interfaces – they include a big number of text and graphical elements; there are situations in which user’s request satisfying supposes introduction of a big data volume; also, the complex interfaces are useful when user needs information represented in different forms in order to assist decision efficiently;
- simple interfaces – they are interfaces that contains a small number of graphical, multimedia and text elements; they are useful when some resources are insufficient or it aims at information representation on a high abstract level.

Design and implementation lead to interface classification as:

- interfaces developed through source code – they suppose use on a programming language for design and functionality assurance of an interface;
- interfaces developed through assistance – they suppose use of an Integrated Development Environment which contains standard controls;
- interfaces generated automatically – in an assistance application for system design, the user supplies details regarding the input and output data, and application generates interfaces on the basis of some standard templates.

Depending on design of information representation zones, the interfaces are:

- structured – there are zones on the display in which information are homogenous for a characteristic;
- non-structured – the interface includes controls and information content arranged randomly.

There is a large diversity of interfaces, classified depending on their characteristics. The informatics application development supposes design and implementation of user interfaces. The developed interface must take into consideration the application objective and it is a part in many classes simultaneously, depending on characteristics that it accomplishes.

3. Interface complexity

Complexity represents the measure in which a system or a component has a architecture or implementation that are hard to be understood and verified, [www].

The complexity level of interfaces is determined by the fallowing aspects:

- the terminal equipments diversity;
- application complexity;
- word diversity in dialog vocabulary;
- speed restrictions for exchange of information development.

Interface complexity regards not only element nature and the links among them but also the volume and difficulty of numerical quantification for interface components, [POPA05].

Informatics complexity and processing kind made by these ones determine an arborescent organization of the interfaces.

The elements number on each level, the height of structure tree, the number of connected elements are primary metrics that contributes to evaluation of *structural complexity* of informatics application interface.

The Solidity and confidence in an interface are determined on the base of ciclomatic complexity, Cx_{CI}:

$$Cx_{CI} = m - n + p$$

where:

m – edge number of the graph associated to interface structure; the graph edges represents the links among the elements within interface structure;

n – element number from interface structure; the application interface is formed from a sub-interfaces order; the interface activation occurs when the user trigs an action of certain type in parent interface;

p – virtual edge number; the virtual edge shows a direct modification of the sub-interfaces at content level.

It is considered the following rface structure :

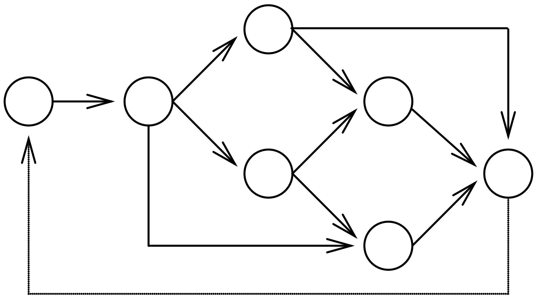

Figure 1 Application interface structure

Regarding the structure depicted in figure 1, it has m = 10 edges, n = 7 sub-interfaces and p = 1 virtual edges. Ciclomatic complexity calculation lead to value $Cx_{CI} = 4$. The indicator value of Cx_{CI} frames the interface from figure 1 in complexity class A with meaning: simple interface, with a low level of complexity. This interface has not high risks when it aims the changing in maintenance.

If it considers the application interface complexity at content level, then it must have in view the following elements:

- base control number used to build the interface, *ncb*;
- relations among controls in terms of link control number, *ncl*;
- link control number among interfaces, *ncs*.

The analytical expression of the model on the base of which it determines the complexity measure is:

$$Cx(DT) = ncb \cdot \log_2 ncb + ncl \cdot \log_2 ncl + ncs \cdot \log_2 ncs$$

The complexity measuring of the relations among sub-interfaces takes place on the base of precedence matrix called MP. Through precedence matrix the dependence, ordering and arrangement of the interfaces are highlighted.

For the structure given in figure 1, the precedence matrix is:

Table no. 1 The Precedence matrix of interface structure

Next sub-interface	*1*	*2*	*3*	*4*	*5*	*6*	*7*

Precedence sub- interface							
1	0	1	0	0	0	0	0
2	0	0	1	1	0	1	0
3	0	0	0	0	0	1	1
4	0	0	0	0	1	1	0
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0

The complexity has the following analytical expression:

$$Cx_M = \sum_{i=1}^{ns}((\sum_{j=1}^{ns} m_{ij})\log_2(\sum_{j=1}^{ns} m_{ij}))$$

where:
ns – sub-interface number from interface structure;
m_{ij} – the element on i row, j column in precedence matrix.
For given matrix, the indicator value is Cx_M = 8,47.
Application interfaces complexity calculation offers important information regarding the operating easiness with informatics application, and maintenance easiness of these ones.

4. Conclusions

The application interfaces are integrant entities in computing resources exploitation by any use. The efficient use of these resources is the result of the interfaces well designed and implemented.
The interface complexity is determined on more levels. Complexity evaluation metrics were defined. The computed values permit the obtaining of conclusions regarding the interfaces quality, using easiness and the maintenance of these ones.

Bibliography

[PRIY04] Saha PRIYATOSH, Cristian TOMA, Ion IVAN – *E-Commerce Project Management*, Economy Informatics, vol. 4, no. 1-4, 2004, pp. 95 – 101

[TOMA04] Cristian TOMA, Ion IVAN – *Mobile Agent Systems and Privacy in Wireless Applications*, Actual Problems of Business Informatics in BRIE Master Program, ASE Printing House, Bucharest, 2004, pp. 102 – 111

[IVAN01] Ion IVAN, Paul POCATILU, Cristian TOMA, Alexandru LEAU – *e3-com*, Economy Informatics, vol. 5, no. 3, 2001, (in Romanian)

[IVAN03] Ion IVAN, Cristian TOMA – *Requirements for Building Distributed Informatics Applications*, The Automatics and Computer Science Romanian Magazine, vol. 13, no. 4, 2003, (in Romanian)

[IVAT03] Ion IVAN, Paul POCATILU, Marius POPA, Cristian TOMA – *The Reliability of M-Applications Based on Transactions*, The Automatics and Computer Science Romanian Magazine, vol. 13, no. 2, 2003, (in Romanian)

[NILS02] Erik NILSSON – *User Interface Modelling and Mobile Applications – Are We Solving Real World Problems*, INFOREC Printing House, Bucharest, 2002, pp. 111 – 119

[HILS99] Alex HILLS – *Wireless Andrew: Bringing Mobile Computing to a University Community of*

10000, IEEE Spectrum, June 1999, pp. 49 – 53

[DUTT99] Amitava DUTTA-ROY – *Fixed Wireless Routes for Internet Access*, IEEE Spectrum, September 1999, pp. 61 – 69

[JING99] Jin JING, Abdelsalam HELAL, Ahmed ELMAGARMID – *Client-Server Computing in Mobile Environments*, ACM Computing Surveys, vol. 31, no. 2, pp. 117 – 157

[POPA05] Marius POPA – *Techniques and Methods for Data Quality Evaluation*, ASE, Bucharest, 2005, PhD Thesis, (in Romanian)

[VARS00] Upkar VARSHNEY, Ron VETTER – *Emerging Mobile and Wireless Networks*, Communications of the ACM, vol. 43, no. 6, June 2000, pp. 73 – 81

[www] http://www.sei.cmu.edu