Jozef Stašák
Ekonomická univerzita Bratislava
Fakulta hospodárskej informatiky
Dolnozemská 1
85 235 Bratislava
Tel. +421/2/67295 5878
+421/2/6729 5856
E- mail: stasak@euba.sk;
jozefstasak@yahoo.com

REPRESENTATION OF INTERFACES VIA FUZZY SET APPARATUS

Summary

This contribution deals with interface problems applied for communication between appropriate application programs. When considering theoretical aspects, the contribution describes a structure and features of such interfaces which is being interpreted via fuzzy sets apparatus, which a linguistic approach is applied for these purposes. As for implementation aspects, the contribution contains an appropriate example and describes principles related to algorithm concerned to design of such interface type.

Keywords

Interfaces, fuzzy sets, linguistic approach, algorithm, implementation principles

ACM classification

B.4 INPUT/OUTPUT AND DATA COMMUNICATIONS, B.4.3 Interconnections (Subsystems), *Interfaces*

JEL classification

M1 – Business Administration, M15 – IT Management

INTRODUCTION

Any information or knowledge based system contains less or more application programs which are required to communicate properly. In most of cases, outputs generated by one application program usually represent inputs for another one; however a content of output sets related to one application program is not usually identical with content of input sets related to another one. As a result of that an appropriate component has to be inserted between both of them. This component is called Interface. The term "Interface - ***Intf***" may be explained as a mechanism by which a user interacts with a given system or an application program and a mechanism that facilitates a user providing input to and/or receiving output from a system. That surface defines a boundary between two application programs as for instance. However, the mechanism supports, facilitates and enforces the syntax, grammar, conversational context, sequence of operations, and semantics rules of the dialog between the parties [5] in the dialog as well. There are more approaches to design and implementation of such interfaces, however a lot of them are usually on J2EE or XML technology applied within web oriented presentation layers [8] where three dimensional interfaces play a role of great importance. Three dimensions and virtual environments intuitively make sense for a wide range of applications, because of the characteristics of the tasks and their match with the characteristics of these environments. Immersion is the feeling of "being there" (replacing the physical environment with the virtual one), which makes sense for applications such as training and simulation. If a user is immersed *and* can interact using natural skills, then the application can take advantage of the fact that the user already has a great deal of knowledge about the world. The immediacy characteristic refers to the fact that there is a short "distance" between a user's action and the system's feedback that shows the result of that action. This can allow users to build up complex mental models of how a simulation works, as for instance [2]. In principle two application programs may also communicate among each other via interface which contains output data set P_{O1} generated by the first application program and several supplementary data P_{O1s} which makes a complete structure of input data set so that the second application program operates properly. The input data set for the second application program proper functionality is denoted as P_{O2}. Therefore there exists an appropriate probability that several P_{O1} and P_{O1s} set elements create an integral part of an interface which provides inputs for a proper communication between both of the above-mentioned application programs. As a result of that, a fuzzy set apparatus may be applied, when providing a quantitative description of the Intf interface. On the other hand, requirements which the set Intf shall meet in order to provide a proper communication between both of the above-mentioned application programs are postulated via logical sentences in a natural language, while the sentences contain thee types of special terms: Principal terms – Pet terms, relating terms Ret -terms and terms to be explained – Tbe terms [6, 7]. This create basis for using of a linguistic approach with the use of fuzzy set apparatus [3, 4, 8].

The contribution deals with theoretical aspects of such approach to interface design and implementation and contains an example such interface structure together with discussion concerned to appropriate implementation aspects.

1 AN INTERFACE – A PRINCIPAL TOOL FOR COMMUNICATION AMONG APPLICATION PROGRAMS

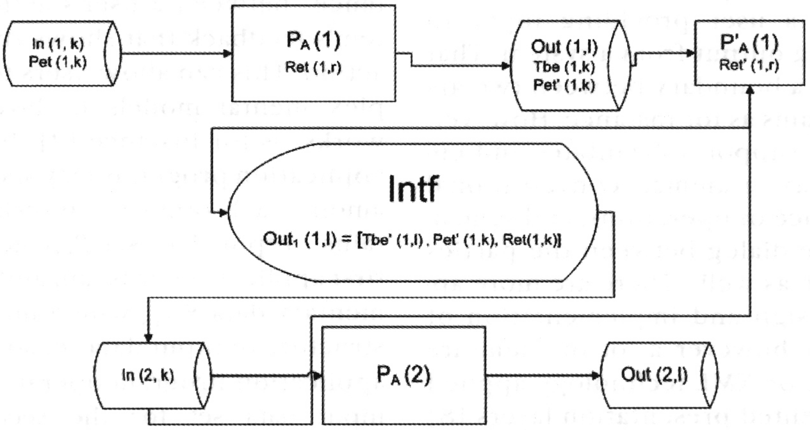
1.1 APPLICATION PROGRAM INTERFACE – STRUCTURE & FEATURES

Information or knowledge based systems contain a set of application programs which have to communicate among each other properly, while each of the application programs provides a conversion of input data into output data with the use of adequate internal functions. However, there are very few cases when outputs generated by one application program may be processed at input of other application program directly. As result of that, a supplementary component has to be inserted between both of application programs. This component is called and Interface and has an adequate structure containing the above mentioned outputs generated by one application program and a set of supplementary elements which represent further necessary criteria X_1(e.g. selection criteria) and elements representing operands or statistic functions X_2. However, there may be further supplementary elements closely related to interface security as well (see also section 3.1). A structure of such interface **(Int)** is shown in Fig. 1 and its position between application programs $P_A(1)$ & $P_A(2)$ is shown in Fig.2. In order to provide a quantitative description of such interface a linguistic approach was chosen together with an appropriate fuzzy set apparatus.

Fig.1 **A structure of Intf interface**

<table>
<tr><td rowspan="2">Output results</td><td>X1</td><td rowspan="2">Supplementary items</td></tr>
<tr><td>X2</td></tr>
</table>

Fig.2 **A position of Intf interface between application programs $P_A(1)$ and $P_A(2)$**



1.2 QUANTITATIVE REPRESENTATION OF INTERFACE STRUCTURE AND FEATURES VIA FUZZY SET APPARATUS

When considering interface structure and features a linguistic approach to interface interpretation was selected and its quantitative representation is based on fuzzy set apparatus, while the following considerations are applied for these purposes:

Consideration no.1 - formulation of requirements which shall provide the application programs $P_A(1)$ & $P_A(2)$

Let us consider two application programs $P_A(1)$ and $P_A(2)$ and appropriate requirements which they shall provide, while the requirements are represented by sentences postulated in a natural language (see also formulas (1 and 2).

$$P_A(1) = R_{eq1} \{V(1,1), V(1,2).....V(1,m_1)] \quad (1)$$

$$P_A(2) = R_{eq2} \{V(2,1), V(2,2).....V(2,m_2)] \quad (2)$$

Each of the above-mentioned sentences consists of adequate terms (Tbe, Pet and Ret [6]) postulated via formulas (3, 4).

$$\forall V(1, i) \in R_{eq1} \exists\ Pet_R (1, i), Ret_R (1, i) \Rightarrow$$
$$\Rightarrow Pet_R (1, i), Ret_R (1, i) \Rightarrow Tbe_R (1, i) \quad (3)$$
for $i=1,... m_1$

$$\forall V(2, i) \in R_{eq2} \exists\ Pet_R (2, i), Ret_R (2, i) \Rightarrow$$
$$Pet_R (2, i), Ret_R (2, i) \Rightarrow Tbe_R (2, i) \quad (4)$$
for $i'=1,... m_1$

On the other hand, the $P_A(1)$ program operates with inputs represented by Pet (1, i) terms and executes functions represented by Ret (1, i'). As a result of that, a set of new values may be created represented by items closely related to Tbe[1] (1, i), Pet' (1, i) and Ret' (1, i) terms[2]. The $P_A(2)$ program operates with inputs Pet (2, j) and executes functions represented by Ret (2, j). As a result of that, a set of new values may be created represented by items closely related to Tbe[3] (2, j), Pet' (2, j) and Ret' (2, j) terms (for j = = 1.... m_2). In most of cases the $P_A(1)$ program outputs do not correspond to $P_A(1)$ program inputs precisely and an appropriate interface shall be inserted between them in order to provide an adequate communication between them (see also Fig.1). As a result of that, an adequate interface structure shall be defined. Therefore, there is a principle question:" *How such interface structure should look like?*" In order to answer this question correctly the next consideration (consideration no.2) shall be postulated.

Consideration No.2 - an interface which plays a role of the principal facility for communication between application program $P_A(1)$ and $P_A(2)$

Let us consider an application program $P_A(1)$ which provides a conversion of input data (represented by Pet (1, i) terms into Tbe (1, i), Pet' (1,i) and Ret' (1, i) terms with use of application program $P_A(1)$ internal functions represented by adequate Ret (1, i) terms. An application program P_A (2) provides similar functions; while its input data are represented by Pet (2, i) terms (see also Consideration no.1). A proper communication between both of the above-mentioned application programs may be represented by formula (5).

$$P_A(1).Int \Rightarrow P_A(2) \quad (5)$$

Consideration No.3

Let us consider two application programs P_{A1} and P_{A2} which shall communicate among each other. Each of the application programs provides appropriate functions which enable converting pre-defined inputs into adequate outputs. This fact may be quantified as follows:

$$\forall P_A (i) \exists\ f_{PA}(i, j), In(i,k) \ \&\ Out (i,l) \Rightarrow$$
$$In(i,k)\ f_{PA}(i, j) \Rightarrow Out (i,l) \quad (5a)$$
for $i = 1.... n(n=2)$, $j = 1...m_1$, $k= 1...m_2$, $l = 1...m_3$ $m_1 \neq m_2 \neq m_3$

When looking at Fig.1, we can see a communication of two application programs P_{A1} and P_{A2} via an appropriate interface Intf, while a set denoted as Out (2,l) consists of two subsets Out_1 (1,l) and Out_2 (1,l) and the set $Out_1 (1,l) \subseteq Intf$ and $Out_2 (1,l) \not\subset Intf$. However, a condition $Intf \equiv In(2, k)$ shall be respected as well. Before searching a solution of a situation represented by formula

1 Tbe terms represent results received based on the operation e.g. Tbe(1,i) = Pet (1,i) Ret(1,i) Pet (1,i+1)

2 The Pet' (1, i) and Ret' (1, i) terms represent output values of the input items denoted as Pet (1,i) and Ret terms. A Ret term may be represented in form of an appropriate query, as for instance. See also example.

3 Tbe terms represent results received based on the operation e.g. Tbe(2,j)= Pet (2,j) Ret(2,j) Pet (2,j+1)

(5), the following auxiliary consideration plays a role of great importance:

Consideration No.4

Let us consider a universal class which is not described strictly and which contains elements closely related to a context represented by functionality of application programs P_{A1} and P_{A2}. As a result of that, a set of generic elements K may be defined which is called "a nuclear space", while the set K consists of two subsets K_1 and K_2. The set K_1 contains elements closely related to functionality of application program P_{A1} and the set K_2 contains elements closely related to functionality of application program P_{A2}. However, these sets create a semantic space E_k closely related to both of the above-mentioned application programs, as well and the following relations may be postulated:

$$K_1 \subseteq K \quad (6)$$

$$K_2 \subseteq K \quad (7)$$

$$E_{k1} \subseteq K_1 \quad (8)$$

$$E_{k2} \subseteq K_2 \quad (9)$$

$$E_k \subseteq K \quad (10)$$

The elements contained in the sets K_1 and K_2 represent the terms in a natural language postulated as follows:

$$E_{k1} \subseteq K_1 = [\ Tbe^1 (i),\ Pet^1 (j),\ Ret^1(k)] \quad (11)$$

$$E_{k2} \subseteq K_2 = [\ Tbe^2 (i),\ Pet^2 (j),\ Ret^2(k)] \quad (12)$$

Where Tbe – means – terms to be explained, Pet – Principal terms and Ret means Relating terms [8].

Consideration No. 5

Let us consider a set $\tau = \{Pet\ (1, i)\}\ (i=1....m1)\}$, while a formula (13) may be postulated

$$\tau = \{Pet\ (1, i)\}_{I'=1}^{m1} = \{In(1, k)\}_{k=1}^{m1} \quad (13)$$

Further, let us consider a set $\rho = \{Ret\ (1,\ i')\}_{I'=1}^{m1}$, while both of the above-mentioned sets t and r are considered to be fuzzy sets defined with respect of relations (6) up to (12). According to assumptions postulated within Consideration no.2, the P_{A1} application program functionality may be represented via semantic relation (14).

$$\{Pet\ (1, i)\}_{I'=1}^{m1} \times \{Ret\ (1, i)\}_{I'=1}^{m1} = s \quad (14)$$

where is represented by relation (15, 16)

$$\sigma = \{Tbe(1, i)\} \otimes \{Pet'(1, i)\}\} \otimes \{Ret'(1, i)\}\} \quad (15)$$

$$\sigma = \tau \times \rho \quad (16)$$

Now, let us look for an appropriate relation between s and In (2, k) (see also Fig.2). With respect o formulas (5, 15 and 16) and Fig.1 a relation (17) may be postulated

$$\sigma \times \lambda = In(2, k) \quad (17)$$

where Tbe(1, i) terms may represent results of computation, as for instance, Pet' (1, i) terms represent output values in form of unchanged Pet terms, it means Pet (1, i) = Pet' (1, i) and Ret' (1, i) terms may represent operands or program functions which enable further processing of Tbe(1, i) and Pet' (1, i) terms e.g. query parameters, statistic functions, etc.

Let us have a look at formula (15) and try to transform it. In order to achieve that we have to postulate the following formulas:

$$\xi = \{Tbe(1, i)\} \otimes \{Pet'(1, i)\}\} \quad (18)$$

$$\psi = \{Ret'(1, i)\}\} \quad (19)$$

$$\sigma = \xi \times \psi \quad (20)$$

With respect of an assumption that the result represented by s is not usually sufficient for a proper communication between application programs P_{A1} and P_{A2} a set $\lambda = \{\lambda(i)\}_{i'=1}^{m1}$ shall be defined and the set contains supplementary elements for a proper functionality of $P_A(2)$ application program.

With respect of relation (5) and assumptions postulated via formulas (21, 22)

$$\forall\ Tbe(1, i'), Pet'(1, i'), Ret'(1,i')\ \exists\ \{\lambda(i')\}_{i'=1}^{m1} \Rightarrow \sigma \times \{\lambda(i')\}_{i'=1}^{m1} = In(2, k) \quad (21)$$

and

$$\forall\ Tbe(1, i'), Pet'(1, i'), Ret'(1,i')\ \exists\ \{\lambda(i')\}_{i'=1}^{m1} \Rightarrow In(1,k)\ .\ Int = In(2,k) \quad (22)$$

the following descriptive fuzzy set may be defined:

$$D(Int) = U\ \sigma <\xi, \lambda>/\psi \quad (23)$$

and this set is called: Descriptive fuzzy set of the interface Int. and the procedure for design of such interface is called Descriptive fuzzy set procedure.

2 DESIGN AND IMPLEMENTATION OF INTERFACES VIA FUZZY SET APPARATUS

2.1 DESIGN OF INTERFACE WITH THE USE OF *DESCRIPTIVE FUZZY SET PROCEDURE*

Algorithm related to design of interface **Int** with the use of *Descriptive fuzzy set procedure* –

Example

Let us consider two application programs $P_A(1)$ and $P_A(2)$. The program PA(2) shall generate an overview concerned to products produced per month, and quarter of year, while the overview shall show a quantity (pieces) of products per month and a quarter of year related to types of products (e.g. displays, processors, keyboards..., etc). The program $P_A(1)$ generates product sums of pieces product total prices per day based on the following inputs (denoted as Pet terms):

Pet (1, 1) = [v_id, hv_dr (e.g. display)]
Pet (1, 2) = [v_id, hv_id (e.g. 23456)]
Pet (1, 3) = [v_nz, hv_nz (e.g. display ABCD)]
Pet (1, 4) = [v_mj, hv_mj (e.g. ks)]
Pet (1, 5) = [v_jc, hv_jc (e.g. 2750)] (24)
Pet (1, 6) = [v_jcm, hv_jcm (e.g. SKK)]
Pet (1, 7) = [v_mn, hv_mn (e.g. 150)]
Pet (1, 8) = [v_dtm, hv_dtm (e.g. 22.8.2007)]

Ret (1, 1) = Pet (1, 5) x Pet (1, 7) (25)

Pet' (1, 1) = [v_dr, hv_dr (e.g. display)]
Pet' (1, 2) = [v_id, hv_id (e.g. 23456)]
Pet' (1, 3) = [v_nz, hv_nz (e.g. display ABCD)]
Pet' (1, 4) = [v_mj, hv_mj (e.g. ks)]
Pet' (1, 5) = [v_jc, hv_jc (e.g. 2750)]
Pet' (1, 6) = [v_jcm, hv_jcm (e.g. SKK)] (26)
Pet' (1, 7) = [v_mn, hv_mn (e.g. 150)]
Tbe (1, 1) = $h_{Ret\ (1,\ 1)}$
Pet' (1, 8) = [v_dtm, hv_dtm (e.g. 22.8.2007)]

X (1, 1) = [v_dr1, hv_dr (e.g display)]
X (1, 2) = [v_dtm1a, hv_dtm1a (e.g. 1.7.2007)]
X (1, 3) = [v_dtm1b, hv_dtm1b (e.g. 1.8.2007)](27)
X (1, 4) = [v_dtm1c, hv_dtm1c (e.g. 31.8.2007)]
X (1, 5) = [v_dtm1d, hv_dtm1d (e.g. 30.9.2007)]

Ret'(1,1) = {select [x(1,1)] & [X (1, 3)] & [X(1,4)] & SUM (Pet'(1,7)] & SUM (Tbe (1,1))
Ret'(1,2) = {select [x(1,1)] & [X (1, 2)] & [X(1,5)] & SUM (Pet'(1,7)] & SUM (Tbe (1,1))

The Int interface consists of three data groups Y1 = Y1 {[Pet' (1, 1).... Pet' (1, 7)]}, Y2=Y2 {[Tbe (1,1), Pet' (1,8)]}, Y3=Y3 {[X(1,1)......X(1,5)]}, Y4= Y4{[Ret'(1,1), Ret (1,2)}.

Int = **Int** (Y1, Y2, Y3, Y4) (29)

Any of the above-mentioned groups may have the following security values:

Any of the above-mentioned groups may have the following security values:

Visible / **In**visible
Public / **Pr**ivate
Protected/**Unpro**tected

However, these levels may be combined as well, e.g. V,Pu, Unpro. In the case no of the security values are written after interface group parameters a combination V,Pu, Unpro is valid and is considered to be a default value.

Int = Int { [Y1 (s1, s2, s3, s4)],
[Y2 (s1, s2, s3, s4)], [Y3 (s1, s2, s3, s4)], (30)
[Y4 (s1, s2, s3, s4)]}

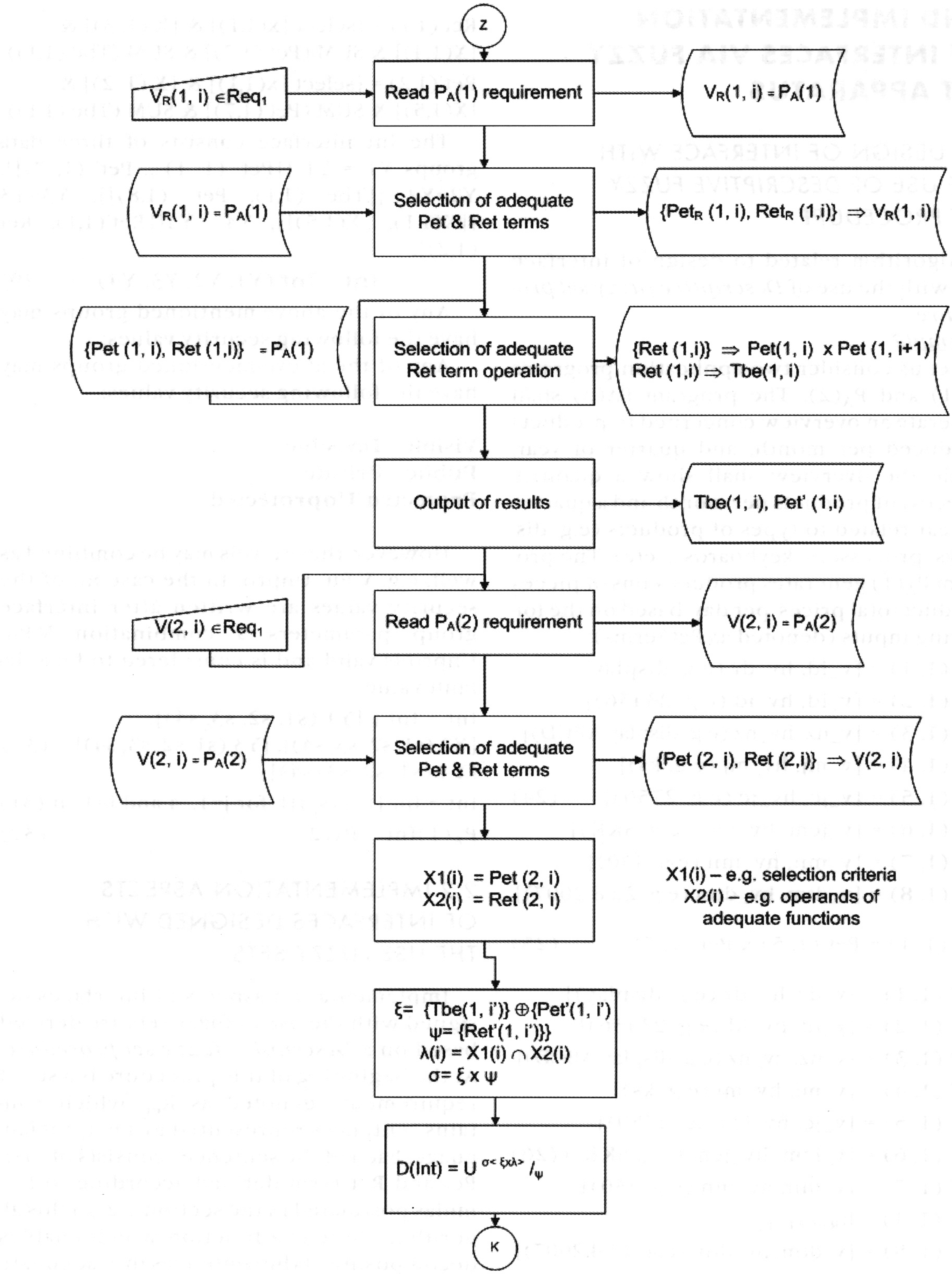
Int = Int {[Y_i (s_j)]} for j=1...4 and i=1...n (31)

P_A(1)Int = P_A(2) (32)

2.2 IMPLEMENTATION ASPECTS OF INTERFACES DESIGNED WITH THE USE FUZZY SETS

Implementation aspects of interfaces designed with the use of fuzzy sets are derived based on a *Descriptive fuzzy set procedure*. At the beginning of this procedure is a set of requirements denoted as R_{eq} which contains sentences represented in a natural language. Each of the sentences consists of Tbe, Pet and Ret term defined according to formulas postulated in the section 2.2. In this algorithm, there is a function which enables decomposing of the sentences into adequate Tbe_R, Pet_R and Ret_R terms. On the other hand, the $P_A(1)$ application program contains an adequate input data represented by Pet (1, i) terms and internal functions represented by appropriate Ret (1, i) terms which

Fig.3 **Basic principles of Descriptive fuzzy set procedure**



enable providing operations according to formula (25), as for instance and receiving results in form of Tbe (1, i) terms and Pet'(1, i). These results are called "Output results" and create the first data set which is an integral part of the interface Int which enables a proper communication between program $P_A(1)$ and $P_A(2)$ (see also formula (32)). In order to provide a proper functionality of the $P_A(2)$ program, it is necessary to assure an adequate structure of inputs denoted as In(2,k) and a semantic analyses of V(2, i) sentences may help to do it. The Pet (2,i) a Ret (2, i) terms are considered to the principal results of the above-mentioned analysis and create bases for subsets X_1 (i) and X_2 (i) Í I (i), while the subset X_1 (i) is closely related to adequate selection and presentation criteria and the subset X_2 (i) is closely related to operands and appropriate statistic functions. With respect of these considerations the following formula may be postulated

$$\lambda(i) = X_1(i) \cap X_2(i) \qquad (33)$$

After that, the description fuzzy set which contains elements closely related to formulas (18, 19, 20 and 23) may be determined. An example of *Descriptive fuzzy set procedure* is described in section 3.1 and the basic principles of the procedure are shown in Fig.3.

CONCLUSION

In this contribution, there are discussed these interface problems which enable a proper communication among actual application programs. Any application program provides a generation of outputs based on appropriate inputs. However, not always a structure of outputs generated by the first program is sufficient for a proper functionality of another application program. Therefore, an adequate interface shall be inserted between them. This contribution contains a structure description related to such interface which is being quantified with the use fuzzy set apparatus. However, there is described the interface example together with design and implementation aspects closely related to this type of interface, as well. Several concrete approaches may be postulated which are closely related application of appropriate packaged application software supporting a system integration. With respect of these approaches, the following methodologies may be postulated: MDIS (Multidimensional methodology for information system design and implementation with the use of appropriate packaged software), ASAP (Methodology concerned to implementation of SAP products) or TISTAS (Methodology for information design and implementation based on adequate packaged software).

The matters described in this contribution create basis for a program development which might enable generating an interface structure elements based on pre-postulated requirements.

REFERENCES

[1] Adamo-Villani, N.- Benes, B.- Brisbin, M.- Hyland, B.: A Natural Interface for Sign Language Mathematics http:// www.envision.purdue.edu//pdfs/ /papers/isvc06a.pdf

[2] Bowman, D. A.: 3D User Interface Design for Virtual Environments http://www.courses.cs.vt.edu/ /˜cs5754/lectures/3DUIs.pdf

[3] Novak, V.: Fuzzy mnoziny a jejich aplikace, Praha: SNTL, 1988.

[4] Ordonez, R.- Zumberge, J. Spooner, J.T. - Passino, K.M.: Adaptive Fuzzy Control: Experiments and Comparative Analyses: http:// www.ece.osu.edu/ /˜passino/PapersToPost/Adapt-fuzzy-exp.pdf

[5] Poelman, L.: Designing) User Interfaces http://www.unicode.org/charts/PDF/U0000.pdf. 34. Greek and Coptic ... cs.nyu.edu/.../lectures/ 01 User Interfaces Overview v1.00 Spring05.pdf

[6] Stašák, J.: A Contribution to Semantic Text Analysis. In: Electronic Computers and Informatics ECI 2004, The University of Technology Košice, Department of Computers and Informatics of FEI, 22-24. 9.2004 Košice - Herľany, SR, p.132-144, ISBN 80-8073-150-0

[7] Stašák, J.: Modeling of Text Semantic with the use of Fuzzy Sets "Economic bulletin of National Technical University of Ukraine "Kiev Polytechnic Institute" 2006

[8] User Interface Development http://www.perfectxml.com/ /manning/ J2EEXMLchap05.pdf

Biography:

Jozef Stašák was graduated at the University of P.J.Šafárik in Košice (1972) and he received his master degree in physics. After having graduated, he concentrated his activities on information science and technology problems and reached his PhD. degree in these branches (1989). At present, he is a university teacher to the Business Informatics Faculty (University of Economy) in Bratislava and his background is closely related to Business Intelligence, Business Process Intelligence problems. He published his own monograph and a co-author more textbooks and studying materials for university students