STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLVII**, Number 2, 2002

A STUDY OF DEPENDENCE OF SOFTWARE ATTRIBUTES USING DATA ANALYSIS TECHNIQUES

MILITON FRENŢIU AND HORIA F. POP

Abstract. The dependence between software attributes is studied, using the projects written by second year students as a requirement in their curriculum. The dendrogram, factorial analysis and principal components methods are used as Data Analysis Technique. Also, some consequences on the education activity are considered.

Keywords: software metrics, measurement, fuzzy clustering, data analysis techniques, education

1. Introduction

The main purpose of Software Metrics is to evaluate the needed resources and to improve the Software development process [6]. Software Metrics are also useful to evaluate the quality of a software product [15]. And, as we show and in this paper, Software Metrics are useful in education. The future programmer will respect an adequate programming methodology if he is thaught to do so.

The dependence between some software product attributes was discussed by many authors [1, 2, 8, 9, 20]. The effect of programming style on some software product attributes was analysed in [7]. Here we consider again this problem, taking in account 29 software attributes and using the Principal Components Method to study the dependence between these attributes.

The fact that code indentation increases software programs readability has been recognized and underlined for a long time [11, 14]. This was later proved by other authors through statistical experiments [16, 17, 19]. Also, it was proved [16] that excessive indentation is useless, that the best result for increasing readability is obtained when 2–4 spaces are used for indentation.

Oman and Cook [17] showed through an experiment that maintenance was performed better by the subjects that had to maintain their programs in book format, than those that had traditional programs. Also, they showed that the

2000 *Mathematics Subject Classification.* 68N30.

1998 *CR Categories and Descriptors.* D.2.3 [**Software**] : Software Engineering – *Coding Tools and Techniques*; I.5.1 [**Computing Methodologies**] : Pattern Recognition – *Models* – *Fuzzy set*; G.3 [**Mathematics of Computing**] : Probability and Statistics – *Data analysis*.

use of typographic style reduces the maintenance effort, improving programmer performance and program comprehension.

There is a close dependence between the clarity, readability and correctness of a program [7]. We all expect that a strong correlation exists between comprehensibility and good design, and this is confirmed again by our experiment. A study of licence examination results based on fuzzy clustering, showing the relationships to programming habitudes is presented in [9].

We have observed that there is a strong dependence between almost all considered attributes. We will try to explain the reason in the Conclusion part of this paper.

2. The experiment

The study is based on 29 projects produced by second year undergraduate students as part of their requirements curriculum. These projects were analysed observing the attributes described in Table 1, and the primary data is given in Table 2.

Attribute	Description	Attribute	Description
A1:	requirements description	A16:	readability
A2:	good specification	A17:	comprehensibility
A3:	function points	A18:	changeability (modifiability)
A4:	design clarity	A19:	structuredness
A5:	design correctness	A20:	testability
A6:	design completeness	A21:	reliability
A7:	design diagrams	A22:	efficiency
A8:	modules specification	A23:	extensibility
A9:	algorithms description	A24:	adaptability
A10:	LOC	A25:	documentation clarity
A11:	no. of comments	A26:	documentation completeness
A12:	good use of comments	A27:	maintainability
A13:	good use of free lines	A28:	simplicity
A14:	indentation	A29:	quality
A15:	good names		

Table 1. Attributes description

The attributes A10 and A11 were measured automatically by computer. All the others were estimated by postgraduate students. All metrics have the values in the interval [0, 10], where 0 stands for "very bad" (or not present at all), and 10 for "excellent". These values are the impressions of the students about the corresponding attributes. Certainly, these values are subjective, but we consider

that this fact does not affect the dependency between the attributes, all values for a project being given by the same person. After all, "subjective measures are cheap and worth using" [5]. Also, we may accept that the postgraduate students are not experienced programmers, but they have finished a similar project three years ago, and another two projects in their third and fourth year. More, half of them are working at Software companies.

These students form a Master group that study the subject "Software Metrics". The definitions of the above considered attributes were given there. These definitions and are inspired from and can be found in [6].

The attribute A12 refers to the documentation done by comments. It is not based on the number of comment lines of the programs. We may write as many comment lines as we like and sometimes the comments contradict the code, or do not reflect what the code does. The measure for this attribute takes in account if the specification of each module is reflected through comments, if the meaning of each variable and object is explained by comments, if the invariants and other important explanations are given by comments.

In [7] a measure for comprehensibility was defined by

$$m(\text{comprehensibility}) = w_1 \cdot m(\text{readability}) + w_2 * m(\text{design}) \tag{1}$$

where $w_1 = 0.4$, and $w_2 = 0.6$, and

$$m(\text{readability}) = [m(\text{comments}) + m(\text{indentation}) + m(\text{names}) + m(\text{spaces})]/4 \tag{2}$$

If we want to verify this hypothesis we may use Chi-square test for our data. For this we must compute the sum

$$\chi^2 = \sum_{i=1}^{n} (c_i - d_i)^2$$

where c_i are the observed values for comprehensibility, and d_i are the computed values using the formula (1). We have considered here that m(design) is the average of m(A3) and m(A7) since the clarity of design and the needed diagrams affect comprehensibility. For 28 degrees of freedom, the critical value of χ^2 at 0.05 level of probability is 41.34. The computed value for our experiment is 7.56, therefore the test is passed.

3. Study of variables dependence

3.1. **Correlation Matrix.** First of all, the correlation coefficients for all pairs (A_i, A_j) were computed[1].

We remark a strong dependence between almost all pairs of attributes. We may observe that the largest correlation coeficients are cor(A3, A10) = 0.98, cor(A2,

[1]Due to space restrictions, the correlation matrix and other results are not printed in this paper. Instead, they are available, together with all other numerical data, on Internet, at the address `http://www.cs.ubbcluj.ro/~mfrentiu/articole/projdat.html`.

Prj	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	6	5	38	5	6	5	6	2	0	759	268	6	8	7	5	6	5	4	4	4	3	5	5	4	6	5	5	6	4
2	6	7	97	6	6	7	5	6	4	2695	22	2	8	5	5	5	5	5	5	4	4	4	5	4	4	5	4	4	4
3	8	8	145	8	8	8	7	9	3	3550	575	7	7	5	5	6	7	5	6	5	5	6	6	6	7	7	6	5	5
4	6	6	58	3	6	5	4	1	0	1600	0	0	5	3	5	4	4	5	5	5	4	5	6	5	5	6	5	6	5
5	8	8	80	7	7	6	7	5	2	2144	130	5	8	8	7	8	7	8	9	8	7	7	7	7	8	7	8	7	7
6	9	9	159	9	8	8	7	6	2	4027	200	6	9	9	5	7	7	7	7	6	7	8	6	7	7	7	6	6	6
7	6	6	61	8	7	7	6	6	2	1649	6	1	9	8	5	5	6	6	8	7	7	9	6	6	7	7	6	6	6
8	9	8	29	7	8	8	6	4	6	653	0	0	6	8	5	5	5	4	4	6	5	4	3	3	5	5	3	5	5
9	9	8	60	7	8	8	7	5	1	1542	0	0	3	8	5	5	7	5	5	5	5	4	6	6	6	6	5	4	5
10	8	9	47	8	7	7	8	8	1	1164	5	1	8	4	5	5	7	5	5	7	5	5	6	7	7	6	5	7	6
11	7	7	90	8	7	7	5	6	2	2816	108	4	7	6	7	7	7	5	7	8	6	6	5	5	8	7	5	7	6
12	8	8	59	7	7	7	6	8	1	1145	68	3	7	5	6	6	6	6	6	5	5	6	6	5	8	7	6	5	5
13	8	8	74	8	8	9	8	9	1	1880	8	1	9	8	7	7	7	7	8	7	7	8	8	7	9	8	7	7	8
14	7	6	68	7	7	8	7	6	5	1680	0	0	7	6	5	5	6	6	7	6	6	7	6	5	8	7	7	6	7
15	8	7	35	7	8	8	6	9	9	715	0	0	4	5	5	4	6	8	7	7	7	8	8	7	8	8	7	8	7
16	6	6	34	6	7	7	6	4	5	780	32	2	6	3	6	5	5	5	5	6	5	5	6	5	7	6	5	6	5
17	9	8	55	8	8	8	8	9	2	1460	30	2	9	7	8	7	8	7	8	8	7	8	7	8	7	7	7	8	8
18	9	8	45	8	8	8	5	8	0	871	0	0	9	9	8	7	6	9	9	8	8	9	8	8	8	7	8	9	9
19	8	9	57	8	9	7	8	8	8	1553	0	0	9	7	8	7	7	6	8	9	8	8	8	8	7	7	8	4	8
20	9	8	49	9	9	8	8	9	6	1289	15	1	9	7	8	7	8	7	9	9	8	9	9	7	8	8	8	9	9
21	9	9	61	9	8	9	8	9	8	1466	4	0	6	7	6	6	7	8	7	9	8	8	8	6	8	7	7	8	8
22	8	7	60	6	7	9	6	6	9	1653	0	0	4	3	8	4	5	5	5	9	8	7	4	4	6	6	5	5	6
23	9	9	86	8	9	9	8	9	7	2105	120	3	8	5	8	7	7	7	7	8	8	8	7	7	9	8	8	7	8
24	8	7	37	7	8	7	7	4	8	752	6	0	9	3	9	6	7	6	6	8	7	8	7	6	9	8	7	7	7
25	8	8	57	7	7	8	8	8	5	1680	192	5	8	7	7	7	8	7	8	6	7	7	7	6	7	8	7	7	7
26	7	6	34	5	6	6	6	2	0	605	38	2	8	7	6	6	6	4	5	4	4	5	5	5	6	7	5	6	6
27	6	6	28	6	7	7	6	3	0	526	6	0	5	5	7	5	5	5	6	6	5	6	5	6	5	5	5	6	5
28	7	6	32	7	7	6	6	3	0	914	0	0	7	6	5	5	5	6	6	6	7	7	7	6	7	7	6	7	6
29	7	7	39	7	8	7	8	7	0	778	153	4	7	7	7	7	7	7	7	6	7	8	7	7	7	7	7	7	7

TABLE 2. The attribute values for 35 projects

A27) = 0.97, cor(A20, 21) = 0.95, cor(A25, 26) = 0.95. The first one is expected, since the size of the product strongly depends of the atribute A3 (function points).

3.2. **Dendrogram.** A dendrogram is a tree that depicts a hierarchical dependence between the attributes, starting from the correlation matrix [12, 13]. Since we refer downwards to this dendrogram, we have drawn it and it can be seen at the above mentioned address (see Figure 1).

3.3. **Factorial Analysis.** A factorial analysis [21] was also performed. The result of this analysis is printed below in Tables 3.a, 3.b and 3.c.

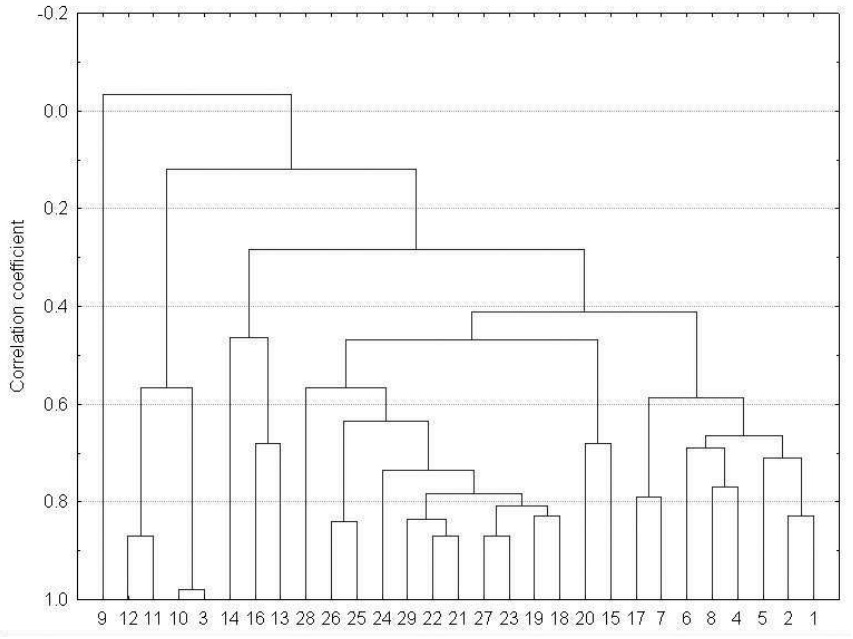


FIGURE 1. Dendrogram for 29 software attributes

To explain the attributes with the probability 0.60 we need two factors, and the relation between attributes and these factors is given in Table 3.a.

Only the attributes A3, A10, A11, and A12 depend more on the second factor than on the first one. All the others depend on the first factor, and we consider this factor the general knowledge of the students. The second "factor" is not a single factor, as can be seen analysing the dendrogram. This dendrogram clearly highlights that A3 and A10 are grouped together, A11 and A12 are also grouped together, but these two groups have quite a low dependency.

We have repeated the factorial analysis for a probability of 0.70 (see Table 3.b). At this level three factors are needed, but, still, the attributes A3, A10, A11, A12, A13, A14 depend stronger on "the second factor". The third factor influences the attributes A1–A8, which characterise the design of the projects.

For the probability 0.80 we need six factors. Now, the second one influences the attributes A3 and A10, and we consider it "the complexity of the problem". The attributes that depend stronger on the third factor are A1, A2, A4, A5, A6, A7, and A17. We can see that the dendrogram classifies these attributes together, in the following subtree: [(((1, 2), 5), ((4, 8), 6)), (7, 17)]. We remark that these

	0.60			0.70				0.80						
Attr.	F1	F2	C	F1	F2	F3	C	F1	F2	F3	F4	F5	F6	C
1	**0.65**	0.24	0.48	0.30	0.12	**-0.76**	0.68	0.21	0.04	**-0.85**	0.11	0.07	0.10	0.79
2	**0.59**	0.44	0.54	0.24	0.32	**-0.77**	0.75	0.15	0.22	**-0.85**	0.12	-0.05	0.13	0.82
3	0.01	**0.86**	0.73	-0.10	**0.82**	-0.30	0.77	0.00	**0.92**	-0.20	0.06	-0.34	-0.05	0.95
4	**0.73**	0.41	0.7	0.47	0.31	**-0.67**	0.76	0.40	0.28	**-0.72**	0.20	0.01	0.16	0.81
5	**0.78**	0.14	0.62	0.45	0.02	**-0.74**	0.75	0.39	0.00	**-0.77**	-0.07	0.08	0.20	0.78
6	**0.57**	0.14	0.34	0.11	0.00	**-0.88**	0.78	0.20	0.27	**-0.76**	-0.26	0.25	-0.04	0.79
7	**0.63**	0.29	0.48	0.44	0.21	**-0.52**	0.51	0.30	-0.26	**-0.73**	0.02	-0.38	0.23	0.79
8	**0.73**	0.33	0.64	0.44	0.23	**-0.71**	0.74	0.49	0.22	**-0.70**	-0.09	-0.09	-0.02	0.78
9	**0.38**	-0.20	0.18	-0.03	-0.31	**-0.70**	0.58	0.11	0.11	-0.42	**-0.80**	0.28	0.16	0.77
10	0.00	**0.81**	0.66	-0.10	**0.78**	-0.29	0.70	-0.02	**0.95**	-0.16	0.04	-0.27	0.00	0.93
11	-0.17	**0.79**	0.65	-0.08	**0.81**	0.06	0.66	-0.06	0.37	-0.04	-0.02	**-0.85**	-0.04	0.82
12	-0.15	**0.84**	0.72	0.03	**0.88**	0.19	0.80	-0.05	0.37	0.05	0.19	**-0.87**	0.15	0.89
13	**0.40**	0.37	0.30	**0.59**	0.39	0.11	0.52	0.32	0.11	-0.02	0.42	-0.28	**0.65**	0.68
14	0.30	**0.37**	0.23	**0.38**	0.37	-0.03	0.29	0.15	0.14	-0.28	**0.94**	0.06	0.09	0.82
15	**0.64**	-0.17	0.44	**0.57**	-0.21	-0.27	0.44	0.35	-0.19	-0.22	-0.26	0.09	**0.86**	0.86
16	**0.56**	0.55	0.62	**0.68**	0.54	-0.08	0.76	0.37	0.09	-0.29	0.47	-0.43	**0.66**	0.93
17	**0.72**	0.48	0.75	**0.60**	0.41	-0.47	0.75	0.44	-0.01	**-0.66**	0.16	-0.42	0.32	0.86
18	**0.83**	0.06	0.70	**0.80**	0.01	-0.31	0.73	**0.86**	0.16	-0.27	0.17	0.17	0.02	0.87
19	**0.84**	0.18	0.73	**0.87**	0.14	-0.22	0.83	**0.81**	0.22	-0.21	0.28	0.09	0.33	0.91
20	**0.80**	-0.20	0.68	0.55	-0.30	**-0.59**	0.73	0.49	0.07	-0.42	-0.28	0.43	**0.51**	0.86
21	**0.90**	-0.06	0.81	**0.67**	-0.15	-0.60	0.83	**0.68**	0.19	-0.44	-0.15	0.36	0.35	0.91
22	**0.86**	-0.01	0.75	**0.83**	-0.06	-0.30	0.79	**0.84**	0.16	-0.19	-0.02	0.19	0.33	0.88
23	**0.82**	-0.02	0.67	**0.84**	-0.05	-0.22	0.75	**0.88**	-0.14	-0.25	0.07	-0.05	0.01	0.86
24	**0.79**	0.15	0.64	**0.79**	0.11	-0.24	0.70	**0.73**	-0.06	-0.35	0.30	-0.07	0.13	0.75
25	**0.77**	0.11	0.61	**0.72**	0.05	-0.33	0.63	**0.75**	-0.02	-0.30	-0.19	-0.18	0.21	0.74
26	**0.78**	0.14	0.63	**0.75**	0.09	-0.31	0.66	**0.82**	0.04	-0.27	-0.18	-0.20	0.15	0.80
27	**0.88**	0.08	0.78	**0.89**	0.04	-0.25	0.86	**0.87**	-0.01	-0.25	0.00	-0.09	0.29	0.90
28	**0.65**	-0.18	0.46	**0.76**	-0.18	-0.01	0.61	**0.75**	-0.24	-0.04	0.11	0.05	0.12	0.64
29	**0.96**	-0.09	0.92	**0.85**	-0.16	-0.43	0.93	**0.76**	-0.10	-0.42	0.06	0.22	0.38	0.94
	(a)			(b)				(c)						

TABLE 3. Factorial analysis for a probability of 0.60, 0.70, and 0.80 respectively

attributes are those connected to the design (similarily to the case of probability 0.70), plus the comprehensibility!

But there are other factors, the fourth one that strongly influences the attribute "indentation", the fifth one, that influences the attributes A11, A12 (comments), and the sixth one influences A13, A15, A16 (mainly the readability of programs).

3.4. **Principal Components Analysis.** Principal Components Analysis (PCA) is designed to reduce the number of variables that need to be considered to a small

number of axes called the principal components, that are linear combinations of the original variables. The new axes lie along the directions of maximum variance thus containing most of the information. PCA provides an objective way of finding attributes of this type so that the variation in the data can be accounted for as concisely as possible. Moreover, due to this space rotation, PCA is often used as a dimensionality reduction method: very few principal components provide a good coverage of all the original variables.

PCA has been applied on the set of variables (i. e. the transpose of the original data set) in order to produce a visual representation of the variables. Table 4 describes the reduction coefficients produced by considering the 29 software attributes and 29 student projects, and Figure 2 presents the scores corresponding to the first two principal components.

No.	Eigenvalue	Successive diff.	Proportion	Cummulative prop.
1	28.8374	28.6766	0.994392	0.994392
2	0.160824	0.160002	0.00554565	0.999938
3	0.000821365	0.000529606	2.83229e-005	0.999966
4	0.000291759	9.32732e-005	1.00606e-005	0.999976
5	0.000198485	4.18986e-005	6.84432e-006	0.999983
6	0.000156587	3.55741e-005	5.39954e-006	0.999989
7	0.000121013	6.58028e-005	4.17285e-006	0.999993

TABLE 4. Reduction coefficients for 29 software attributes and 29 student projects (the remaining eigenvalues are less than 0.0001, and less important)

From an analysis of Figure 2 we remark the three isolated points (corresponding to software attributes A3, A10 and A11). Because of the agglomeration of points in the remaining region, we will repeat the procedure, but this time will ignore the three above mentioned attributes. The results are depicted in Table 5 and Figure 3.

3.5. **Fuzzy Hierarchic Clustering.** The theory of fuzzy sets was introduced in 1965 by Lotfi A. Zadeh [22] as a natural generalization of the classical set concept. Let X be a data set, composed of n data items characterized by the values of s characteristics. A fuzzy set on X is a mapping $A : X \to [0, 1]$. The value $A(x)$ represents the membership degree of the data item $x \in X$ to the class A. The advantage of this approach is that it allows a data item x to be a member of more classes, with different membership degrees, according to certain similarity criteria.

Clustering algorithms based on fuzzy sets have proved their superiority due to their ability to deal with imprecise sets, imprecisely-defined boundaries, isolated points, and other delicate situations. The class of fuzzy clustering algorithms based

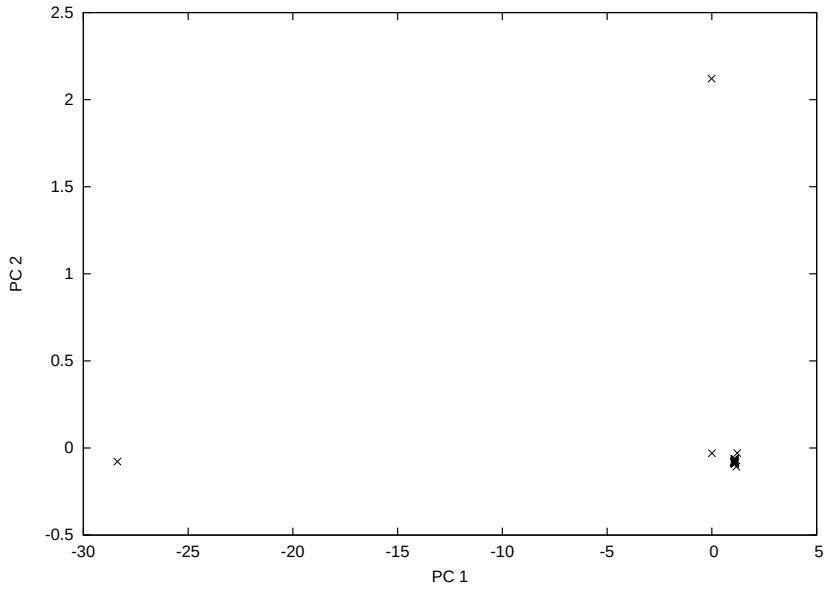


FIGURE 2. Representation of scores corresponding to PC1 and PC2 for 29 software attributes

on fuzzy objective functions [3] provides a large share of geometrical prototypes and combinations thereof, to be used according to the data substructure. On the other hand, the Fuzzy Divisive Hierarchical scheme [4, 18] provides an in-depth analysis of the data set, by deciding on the optimal subcluster cardinality and the optimal cluster substructure of the data set.

The visual representations in Figures 2 and, especially, 3 enable us to further analyse the attributes set. Due to the obvious linear structure of the data, we consider the Fuzzy Divisive Hierarchic Clustering (FDHC) algorithm with linear prototypes. In order to fully understand the relationships between the software attributes, we have used both the original set of 29 software attributes, as well as the smaller set of 26 attributes, where attributes A3, A10 and A11 have been ommitted.

The classification tree and the final binary partition produced by FDHC with linear prototypes using the set of 29 normalized attributes are depicted in Figure 4.

The classification tree and the final binary partition produced by FDHC with linear prototypes using the set of 26 normalized attributes are depicted in Figure 5. The corresponding fuzzy membership degrees to the classes from the final fuzzy partition are displayed in Table 6.

No.	Eigenvalue	Successive diff.	Proportion	Cummulative prop.
1	15.77	11.0031	0.543793	0.543793
2	4.7669	2.66606	0.164376	0.708169
3	2.10084	0.720066	0.0724429	0.780612
4	1.38078	0.34621	0.047613	0.828225
5	1.03457	0.272993	0.0356747	0.863899
6	0.761573	0.168301	0.0262612	0.890161
7	0.593272	0.090646	0.0204577	0.910618
8	0.502626	0.109051	0.0173319	0.92795
9	0.393575	0.0607744	0.0135716	0.941522
10	0.332801	0.0378924	0.0114759	0.952998
11	0.294909	0.0311804	0.0101693	0.963167
12	0.263728	0.0901184	0.00909408	0.972261
13	0.17361	0.00443054	0.00598655	0.978247
14	0.169179	0.0390419	0.00583377	0.984081
15	0.130137	0.0416428	0.0044875	0.988569
16	0.0884946	0.0165013	0.00305154	0.99162
17	0.0719933	0.0172449	0.00248253	0.994103
18	0.0547483	0.0127307	0.00188787	0.995991
19	0.0420176	0.0133489	0.00144888	0.99744
20	0.0286687	0.00901874	0.000988575	0.998428
21	0.0196499	0.00936073	0.000677584	0.999106
22	0.0102892	0.00124848	0.0003548	0.999461
23	0.00904073	0.00496486	0.000311749	0.999772
24	0.00407587	0.00154728	0.000140547	0.999913
25	0.00252859	0.00252859	8.71928e-005	1
26	2.5631e-016	1.55938e-016	8.83826e-018	1
27	1.00372e-016	6.85124e-016	3.4611e-018	1
28	-5.84752e-016	9.65261e-017	-2.01639e-017	1
29	-6.81278e-016	-6.81278e-016	-2.34923e-017	1

TABLE 5. Reduction coefficients for 26 software attributes and 29 student projects

By analysing the Figures 4 and 5, we remark that the cluster substructure of the set of attributes became more detailed after the three isolated attributes (A3, A10, A11) have been removed. This is consistent with the preceding remark on the quality of the PCA projection obtained without the same three atttributes, as we can see from Figures 2 and 3.

A completely different remark may be drawn by analysing Table 6. We see there that many of the attributes have very close fuzzy membership degrees to the four

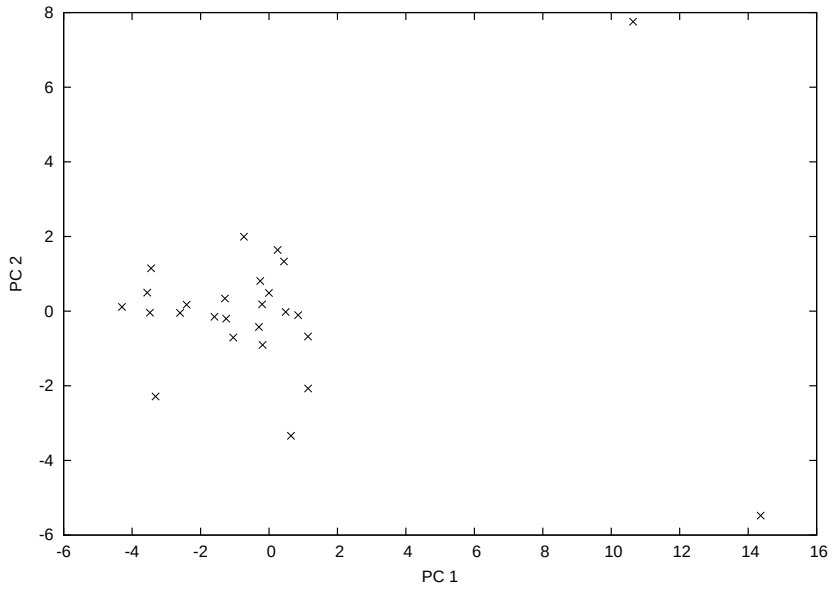
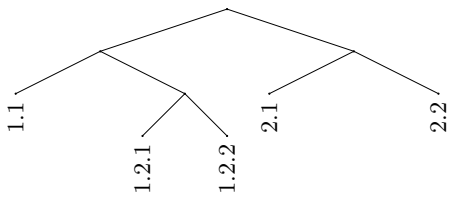


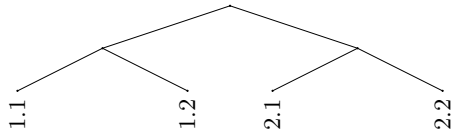
FIGURE 3. Representation of scores corresponding to PC1 and PC2 for 26 software attributes



Class	Members
1.1.	1 2 4 5 6 7 12 13 14 15 16 17
1.2.1.	3 18 19 22 23 24 25 26 27
1.2.2.	8 9 20 21 28 29
2.1.	11
2.2.	10

FIGURE 4. Classification tree and final partition for the set of 29 normalized attributes

final classes. More specifically, 10 attributes, out of the total of 26 (A7, A8, A14, A15, A20, A23, A24, A25, A26, A28), have dominant fuzzy memberships between 0.25 and 0.35. Because of this very strong fuzziness, these attributes should be considered as being shared by all the four classes. They do not contribute to shaping the cluster substructure, and, effectively, will be associated to a fifth class, a sort of 'unclassified' class. Out of the remaining 16 attributes, ten have membership degrees between 0.30 and 0.50 (A4, A5, A6, A17, A18, A19, A21, A22, A27, A29), four have membership degrees between 0.50 and 0.80 (A1, A2, A13, A16) and only two have membership degrees between 0.80 and 1.00 (A9,



Class	Members
1.1.	8 9 15 19 20 21 22 28
1.2.	13 18 23 24 27 29
2.1.	1 2 5 6
2.2.	4 7 12 14 16 17 25 26

FIGURE 5. Classification tree and final partition for the set of 26 normalized attributes (without A3, A10 and A11)

Attr.	1.1.	1.2.	2.1.	2.2.
1	0.0975649	0.0962468	**0.640445**	0.165744
2	0.110721	0.130625	**0.559055**	0.199599
4	0.146171	0.165414	0.336343	**0.352072**
5	0.0959909	0.100097	**0.455519**	0.348393
6	0.112062	0.121853	**0.49245**	0.273635
7	0.175628	0.206314	0.297583	**0.320475**
8	**0.254044**	0.246184	0.249927	0.249846
9	**0.997947**	4.30E-05	0.000966477	0.0010432
12	0.00224497	0.00176989	0.000225756	**0.995759**
13	0.0122362	**0.513947**	0.217884	0.255933
14	0.219962	0.228581	0.273131	**0.278326**
15	**0.284285**	0.278515	0.229546	0.207654
16	0.12194	0.137694	0.238524	**0.501841**
17	0.148125	0.168134	0.257769	**0.425972**
18	0.278445	**0.362585**	0.244657	0.114314
19	**0.384798**	0.308419	0.179179	0.127605
20	**0.315049**	0.292766	0.211903	0.180283
21	**0.380813**	0.378577	0.1516	0.0890098
22	**0.376193**	0.313267	0.177344	0.133196
23	0.30867	**0.347824**	0.22084	0.122666
24	0.239836	**0.29741**	0.3165	0.146254
25	0.251962	0.22761	0.244006	**0.276423**
26	0.226493	0.244955	0.259432	**0.26912**
27	0.323874	**0.381824**	0.207184	0.0871175
28	**0.297997**	0.295768	0.233317	0.172919
29	0.377936	**0.400709**	0.149011	0.0723431

TABLE 6. Fuzzy membership degrees to the final partition for the set of 26 normalized attributes (boldfaces indicate the membership degree to the major defuzzified class)

A12). These last 16 attributes form, actually, the core of the data substructure. The final partition, modified as described, is presented in Table 7.

Class	Members
1.1.	9 19 21 22
1.2.	13 18 27 29
2.1.	1 2 5 6
2.2.	4 12 16 17
(unclassified)	7 8 14 15 20
	23 24 25 26 28

TABLE 7. Final partition for the set of 26 normalized attributes, modified by isolating the attributes with dominant fuzzy membership degrees between 0.25 and 0.35

We also conclude that our fuzzy clustering analysis shows three different kinds of attributes, based on the fuzzy membership distributions. The first main set of attributes is formed by the three strongest attributes removed in the first instance and the two attributes with very large membership degrees (A3, A9, A10, A11, A12). These attributes, also forming the core of the 29 attributes classification (as we see from Figure 4), are the best separated attributes. A second set of attributes correspond to the attributes having resonably high fuzzy membership degrees, and is formed by the attributes from classes 1.1, 1.2, 2.1 and 2.2 from Table 7, other than A9 and A12. These attributes have been classified with a high degree of certainty, but are not as crisp as those in the first set. Finally, the set of 'unclassified' attributes, as presented in Table 7, have fuzzy membership degrees distributed almost evenly between the four fuzzy classes, suggesting that they are not quite suitable to help discriminating among the analysed set of student projects.

4. Concluding remarks

As we expected, very few correlation coefficients are close to zero, and this shows that there is a strong dependence between almost all attributes. As the factorial analysis proved, with the exception of the three above mentioned attributes, plus A9 and A12, these attributes are dependent on the general knowledge of the programmers, which is the main factor that influences all attributes. But there are other factors, and the factorial analysis revealed the complexity of the solved problems and the "discipline", i. e. the wish and habituation of respecting the methodology of programming; it is not sufficient to know it, we must respect it. We may conclude that a good programming style and a correct programming habit must be taught in parallel.

Also, we must observe some anomalies, and, for educational purposes, take some measure to eliminate them. First, we can observe that the smallest coefficients correspond to the comments (second column), and one factor of factorial analysis

is strongly connected to this attribute. By analysing the primary data, we may observe that students do not like writing comments (9 projects out of 29 have no comments at all)! We all know that software documentation is generally poor, often the only information we have is the source code, and we see the reason.

As was observed earlier [9], "the indentation rules are much better obeyed. There is one more reason for this. At all lectures, when the teachers write algorithms or code, they respect these rules in all lines. But only sometimes they write comments." But, also, we must observe a progress: the students are more aware than one year ago [7] that they must write comments in their programs.

We have analysed Software products made by undergraduate students. We are confident that the results cannot be extrapolated to large software systems, but they may be used to provide better instruction for the students, and may be used as effective didactic materials, especially for the course on "Software Metrics". Even if at the level of the first year of study we insist on the necessity of a personal style in programming, and of fulfilling a series of important rules [7, 8], the students are skeptical, they are happy that their programs "work". They do not like to write comments, or to insist on a good design and Pseudocode algorithms, or documentation.

The masters students have seen completely differently the necessity to have a full and correct documentation, its usefulness, the effect of an adequate programming style on the final software products.

By studying the primary data obtained by the masters students we have observed for four projects large discordances between attributes A3 (function points) and A10 (size). By making a more careful analysis of these projects we noticed that they are actually incomplete, since they have only fully implemented a part of the functions required at the specification phase. By not being finalised projects, they have been eliminated from the subsequent data manipulation phases. But this fact in itself underlined another usefulness of assesment of these attributes, as well as the necessity of having some adequate tools for student projects assesment.

Acknowledgements

We would like to acknowledge the support of postgraduate students of the group "Component Based Programming" for their help in analysing the projects.

References

[1] Ronald Baecker, Aaron Marcus, *Design Principles for the Enhanced Presentation of Computer Program Source Text*, CHI'86 Proceedings, 51–58.

[2] Victor R. Basili, Richard W. Selby, David H. Hutchens, *Experimentation in Software Engineering*, IEEE Transactions on Software Engineering, Vol. Se-12 (1986), no.7, 733–743.

[3] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[4] D. Dumitrescu, *Hierarchical pattern classification*, Fuzzy Sets and Systems 28 (1988), 145–162.

[5] A. Dunsmore, M. Roper, *A Comparative Evaluation of Program Comprehension Measures*, EfoCS-35-2000, Department of Computer Science, University of Strathelgde, Glasgow, 2000.
[6] N.E. Fenton, *Software Metrics*. A Rigorous Approach, Int. Thompson Computer Press, London, 1995.
[7] M. Frenţiu, *On programming style – program correctness relation*, Studia Univ. Babes-Bolyai, Series Informatica 45, 2 (2000), 60–66.
[8] M. Frenţiu, *The Impact of Style on Program Comprehensibility*, "Babeş-Bolyai" University of Cluj-Napoca, Research Seminar on Computer Science, 2002, pp. 7–12
[9] M. Frenţiu, H. F. Pop, *A study of licence examination results using Fuzzy Clustering techniques*, Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Research Seminars, Seminar on Computer Science, 2001, 99–106.
[10] Don Gesink, *Software Metrics – The Art of Measuring Software Quality*, 1995, `http://www.interex.org/pubcontent/interact/apr95/pp56.html` and `http://www.interex.org/pubcontent/interact/oct95/05softqu/softqu.html`
[11] D. Gries, *The Science of Programming*, Springer Verlag, Berlin, 1981.
[12] John W. Harbaugh, Daniel F.Merriam, *Computer applications in stratigrafic analysis*, John Wiley & Sons, NewYork, 1968.
[13] G. Lazarovici, M. Frenţiu, *Methods for Automated Classification Use in Archaeology. An Application to Neolithic Graves and Ornaments*, First Romanian Conference on the Application of Physics Methods in Archaeology, Cluj-Napoca, 5–6 November 1987.
[14] H.F. Ledgard, *Programming Proverbs for Fortran Programers*, Hayden Book Company, Inc., New Jersey, 1975.
[15] Steve McConnell, *Software Quality at Top Speed, Software Development*, 1996, `http://www.construx.com/stevemcc`
[16] Richard J. Miara, Joyce A. Musselman, Juan A. Navarro, and Ben Shneiderman, *Program Indentation and Comprehensibility*, Comm. A.C.M., 26 (1983), no.2, 861–867.
[17] Paul W. Oman and Curtis R. Cook, *Typographic Style is More than Cosmetic*, Comm.A.C.M., 33 (1990), no. 4, 506–520.
[18] H.F. Pop, *SAADI: Software for fuzzy clustering and related fields*, Studia Universitatis Babeş-Bolyai, Series Informatica 41, 1 (1996), 69–80.
[19] Armstrong A. Takang, Penny A. Grubb and Robert D.Macredie, *The effects of comments and identifier names on program comprehensibility: an experimental investigation*, Journal of Programming Languages, vol. 4 (1996), 143–167.
[20] Iris Vessey, Ron Weber, *Some Factors Affecting Program Repair Maintenance: An Empirical Study*, Comm. A.C.M., 26 (1983), no. 2, 128–134.
[21] S.Watanabe, H.Haven, *Karhunen-Loeve expansion and Factor analysis. Theoretical remarks and applications*, in Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes, Prague 1967.
[22] L.A. Zadeh, *Fuzzy sets*, Information and Control 8 (1965), 338–353.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, RO-3400 Cluj-Napoca, Romania
E-mail address: `mfrentiu@cs.ubbcluj.ro, hfpop@cs.ubbcluj.ro`