Visualization and data mining method for inventory classification

Innar Liiv, *Member*, IEEE

***Abstract*—Inventory classification of stock-keeping units is typically achieved with ABC analysis accompanied with a diagram describing the distribution of dollar-usage values. Unfortunately, it fails to identify the interdependencies between the products, which may lead to alienating customers by ignoring the effect of assortments of choices. To remedy this problem, we propose new visualization and product interdependency identification methods. Experimental results in real-world scenarios for two warehouse datasets are included and analyzed.**

***Index Terms*— inventory management, data mining, ABC classification, annual-dollar-usage ranking method.**

I. Introduction

All the individual items which comprise the total inventory are not of equal relative importance, therefore stock-keeping units (SKUs) are commonly grouped together and generic inventory stock control policies are applied for each group. Typically, Pareto's Principle (80/20) is used for classifying and prioritizing items, which is called the annual dollar usage ranking method [1].

However, the situation of a product being frequently bought, assembled or used together with some other product is often disregarded. Ignoring such behaviour may lead to customer retention for those who are accustomed of buying specific products in bundles. Rust *et al.* [17] described a similar effect as *Profitable Product Death Spiral*, "in which decisions that seem to be increasing profitability alienate the customer by ignoring the effect of assortments of choices, eventually leading the firm to disaster". Rust *et al.* suggest conducting focus-group interviews to determine those products that have interdependencies. In this paper, we suggest methods using transaction history records available in inventory management software. The visualization approach is related to group technology methods in manufacturing (i.e. machine-component cell formation), but the goal is rather to establish a two-mode typicality scale, not to form blocks near diagonal.

I.Liiv is with the Department of Informatics, Tallinn University of Technology, 15 Raja Street, EE-12618 Tallinn, ESTONIA (phone: +3725200552; fax: +3726202305; e-mail: innar.liiv@ttu.ee).

This work was supported in part by the Estonian Information Technology Foundation under Grant 06-03-99-08.

1-4244-1118-1/07/$25.00 ©2007 IEEE.

The main contributions of this paper can be summarized as follows:

- We propose two new algorithms for ranking and grouping stock keeping units and transactions according to customer behaviour;
- We propose a better procedure for *demand association conflict* detection than [13];
- We present a visualization method for inventory transaction history records where product interrelations can be perceived;
- We evaluate the above methods with real-world datasets from [13].

We will give references to related work in inventory classification and group technology in manufacturing (in section 2) and propose a class of visualization and data mining methods (in section 3). Experimental results in real-world scenarios are given in sections 4, followed by the conclusion.

II. Related work

We will divide the related work into two paragraphs, where we will very shortly summarize the advances in ABC inventory classification (see [13] for an extended recent review) and group technology in manufacturing.

Inventory ABC classification. H.F. Dickie from General Electric Company was the first to apply Pareto's principle in inventories [1], he called it the "ABC Inventory Analysis" and presented several successful implementations from different departments to emphasize that managerial attention allocated to items should be in proportion to their importance. Later, there was a lot of discussion criticism towards the use of single criterion [2-12], presenting enhanced methods based on joint criteria matrix [2,3], Saaty's Analytic Hierarchy Process (AHP) [5], genetic algorithms [6,7] and neural networks [9].

The methodology presented in this paper is similar to [12], which clustered the items based on operational attributes about each item. Current paper also uses the unsupervised learning paradigm, but performs clustering of the inventory transactions, therefore segmenting items based on transaction and movement behaviour. The approach presented in this paper is an extension to methodology presented in [13], where the main emphasis was on identifying the demand association conflicts (induced rules, which detected

```
PROCEDURE CALCULATE_CONFORMITY() // All variables except I;J;CONFORMITY are global
  FOR I=0 TO number of rows
    FOR J=0 TO number of elements in row I
      IF DATA[I][J] NOT in the EXCLUDED array THEN
        CONFORMITY[DATA[I][J]]=CONFORMITY[DATA[I][J]]+(number of non-excluded elements on row I)
      END IF
    NEXT J
  NEXT I
  FOR EACH (element,frequency) in FREQ
    IF FREQ.element NOT in array EXCLUDED THEN
      CONFORMITY[FREQ.element]=
              = 2*CONFORMITY[FREQ.element]+(B-A)-(FREQ.frequency*(total number of non-excluded items))
    ENDIF
  END
  RETURN CONFORMITY
END PROCEDURE

A = total number of elements
B = number of items (SKU) * number of rows (transactions)
FREQ is frequency table array for all items (index: name, value: frequency)
DATA is an array for transactions and items within it, read directly from the input file

FOR I=0 TO number of items
  FIND MINIMAL ELEMENT MIN_ELEMENT FROM RESULTS OF CALCULATE_CONFORMITY()
  IF PREVIOUS.value < MIN_ELEMENT.value PRINT "--- END OF GROUP ---"
  PRINT MIN_ELEMENT.name AND MIN_ELEMENT.value
  PREVIOUS.value=MIN_ELEMENT.value
  A = A - FREQ[MIN_ELEMENT.name]
  B = B - (number of rows)
  ADD MIN_ELEMENT.name to array EXCLUDED
NEXT I
```

Figure 1. Algorithm for ranking and grouping stock keeping units according to customer behaviour

interdependencies between the products from different ABC classes). Also, as the method in [13] was based on the association rule framework [14-16], it required to choose confidency level based on managerial judgement, where trial-and-error is probably the only search strategy for finding those thresholds. In this paper we present a method, which requires no parameter configuration.

Group Technology in manufacturing. Methods similar to those presented in this paper, are often applied in manufacturing under the name "group technology" or "product flow analysis". However, their aim is to permutate matrices to form blocks in the diagonal, i.e. result a block-diagonal seriation.

The general idea of product flow analysis to classify the components into product families was introduced already by Burbidge in [18] and independently by Mitrofanov [20], the machine/part matrix was presented in [19]. The results were obtained manually, first attempt to develop a non-intuitive algorithm was by McAuley [21], who also stated that "at present, as far as is known, the only way of finding the groups of machines and families of parts is to rearrange the rows and columns of the matrix, by hand, until the pattern [...] is obtained". McAuley's solution was based on the works of Sokal [22] and Kendall [23]. After those works, more computerizable procedures were published (e.g. [25-27]).

A comprehensive overview of the research issues with applicability and justification discussion in cellular manufacturing is available in [24]. For discussion on seriation methods in general, an extended recent review is available in [32].

III. PROPOSED METHODOLOGY

We present an enhanced inventory classification methodology for detection and visualization of interrelations between the stock keeping units, especially from different ABC-classes.

To reach that goal, we first propose two new enhanced algorithms for ranking and grouping stock keeping units (see Figure 1 for pseudocode of the algorithm) and transactions (see Figure 2) according to customer behaviour. Both algorithms are based on the data analysis methods presented by Vyhandu ([28]-[30]). However, all of those algorithms worked with full two-mode data arrays, which resulted a serious overhead with sparse matrices. Trivial change in data structures (e.g. using linked lists) would not help due to algorithms' restrictions - non-occurrences had to be also considered. Besides the memory overhead, running those algorithms was not feasible with SKU > 1000 and number of transactions > 10000. Such dimensions are unfortunately very common in most of the transaction history records even with small companies. Therefore, enhanced algorithms were developed to meet the requirements of real-world scenarios.

The result of the proposed algorithms is an ordered list of the stock keeping units and transactions according to the typicality in the buying behaviour. Similarly to group technology in manufacturing, it is also useful to reorder such adjacency matrix of transactions and stock keeping units according to the new obtained order to understand the natural organization of the data. Examples of visualizing such binary matrix with dot plotting are available on figures 5 and 6.

```
01:PROCEDURE CALCULATE_CONFORMITY() // ALL variables except I;J;CONFORMITY are global variables
02:  FOR I=0 TO number of rows
03:    IF I NOT in EXCLUDED THEN
04:      FOR J=0 TO number of elements in row I
05:        CONFORMITY[I]=CONFORMITY[I]+FREQ[DATA[I][J]]
06:      NEXT J
07:      CONFORMITY[I]=2*CONFORMITY[I]+(B-A)-(number of not excluded rows)*(number of elements on row I)
08:    ENDIF
09:  NEXT I
10:  RETURN CONFORMITY
11:END PROCEDURE

A = total number of elements
B = number of items (SKU) * number of rows (transactions)
FREQ is frequency table array for all items (index: name, value: frequency)
DATA is an array for transactions and items within it, read directly from the input file

12:FOR I=0 TO number of rows
13:  FIND MINIMAL ELEMENT MIN_ELEMENT FROM RESULTS OF CALCULATE_CONFORMITY()
14:  IF PREVIOUS.value < MIN_ELEMENT.value PRINT "--- END OF GROUP ---"
15:  PRINT MIN_ELEMENT.name AND MIN_ELEMENT.value
16:  PREVIOUS.value=MIN_ELEMENT.value
17:  A = A - FREQ[MIN_ELEMENT.name]
18:  for ($j=0;$j<count($data[$min[0]]);$j++) { $freq[$data[$min[0]][$j]]--; $A--; }
19:  FOR J=0 TO number of elements in row MIN_ELEMENT.name
20:    FREQ[DATA[MIN_ELEMENT.name][J]]--
21:    A--
22:  NEXT J
23:  B = B - FREQ.Count()
24:  ADD MIN_ELEMENT.name to array EXCLUDED
25:NEXT I
```

Figure 2. Algorithm for ranking and grouping inventory transactions according to customer behaviour

A. Numerical Example

For consistency and comparability reasons, we will use example dataset from [13]. Table 1 shows four items (stock keeping units) referred to as i_1 to i_4. Each transaction t is represented as a binary vector, with $t[k]=1 \leftrightarrow i_k \in t$ if i_k was bought, assembled or used in transaction t. The quantity of each item in the transaction history record is ignored, as we are concerned about the association. *DollarValue* of an item (in the last row) is the result of the classical ABC analysis, which is calculated independently.

TABLE 1
TRANSACTIONS AND DOLLAR-USAGE VALUES

	i_1	i_2	i_3	i_4
t_1	1	0	0	1
t_2	0	1	0	0
t_3	0	0	1	0
t_4	0	0	1	0
t_5	0	1	0	0
t_6	1	0	0	1
DollarValue	36	6	1	1

```
Item1 Item4
Item2
Item3
Item3
Item2
Item1 Item4
```

Figure 3. Data from Table 1 in transactional format

After applying the algorithm presented on Figure 1 (data has to be in the format presented on Figure 3), we will get the following result (new order for items):

```
1. Element I3; Weight 12
2. Element I2; Weight 10
----- END OF GROUP -----
3. Element I4; Weight 12
4. Element I1; Weight 6
```

Secondly, when we apply algorithm on Figure 2, we will get the following result (new order for transactions):

```
1. Element T6; Weight 12
2. Element T1; Weight 8
----- END OF GROUP -----
3. Element T5; Weight 12
4. Element T2; Weight 8
5. Element T4; Weight 8
6. Element T3; Weight 4
```

If we sort the matrix according to the new orders obtained, we get matrix presented in Table 2. Sometimes it is reasonable for efficiency purposes to calculate only the conformity weights (procedures `CALCULATE_CONFORMITY` in both algorithms) for the objects instead of performing full step-by-step iterations (lines 17-25 and 12-25 in algorithms). For our numerical example, the conformity (typicality) weights would be the following:

- i_1 (16), i_4 (16), i_2 (12), i_3 (12);
- t_2 (14), t_3 (14), t_4 (14), t_5 (14), t_6 (12), t_1 (12).

One can see that rankings according conformity weights and the order obtained by the whole algorithm are similar. If such results satisfy, it enables to save a lot of computational time. However, plain conformity calculation does not provide comparable neighbouring similarity maximization nor the identification of cluster boundaries.

TABLE 2
PREVIOUS TABLE AFTER REORDERING

	i_1	i_4	i_2	i_3	*Influence weight*
t_6	1	1	0	0	12
t_1	1	1	0	0	8
t_5	0	0	1	0	12
t_2	0	0	1	0	8
t_4	0	0	0	1	8
t_3	0	0	0	1	4
DollarValue	36	1	6	1	
Influence weight	6	12	10	12	

From this example we also notice an interesting property of the algorithms. Only two groups of transactions were detected in the dataset, although using visual investigation, we can identify three. Such difference only emerges with very small datasets and it actually demonstrates that there are only two equally balanced groups with regards to the significance.

Another important distinction to such inventory classification methodology is that we also get the transaction segmentation (transition description from typical inventory transactions to untypical) and understanding of the inner structure of the item-transaction co-behaviour.

From the results of the ranking of the items, demand association conflicts are easily detectable. Let Q denote the new order for transforming Table 1 into Table 2. Then demand association conflicts with n-th element can be detected with the following conditions:

Strong conflict:
Class(Q(n-1))=A *and* Class(Q(n))=C *and* Class(Q(n+1))=A
Conflicts:
Class(Q(n-1))=A *and* Class(Q(n))=B *and* Class(Q(n+1))=A
Class(Q(n-1))=B *and* Class(Q(n))=C *and* Class(Q(n+1))=B
Class(Q(n-1))=A *and* Class(Q(n))=C *and* Class(Q(n+1))=B

Conflicts between classes are thus obtained from ABC-classification and de facto customer behaviour - products from different ABC classes that have strong interdependencies.

As mentioned before, it is also possible to perform a quick and non-exhaustive ranking using the algorithms on Figure 1 and Figure 2 - instead of complex nearest neighbour search, new order can be obtained directly from the first calculation of conformity procedure. However, disregarding the rather similar visual output, detection of the demand association conflicts performs much worse with plain conformity calculation.

For feasibility and practicality reasons, instead of matrices with numerical values, dot plotting (dot denoting value=1 in the matrix) is used with larger datasets.

IV. EXPERIMENTAL RESULTS

The aim of the experiments is to evaluate the applicability and practicality of the presented methods in real world scenarios. We will use wholesale company inventory transaction data from [13]. The number of SKUs in the datasets is 234 (Dataset 1) and 1601 (Dataset 2), respectively. Distribution of dollar-usage values in datasets can be seen on Figure 4.

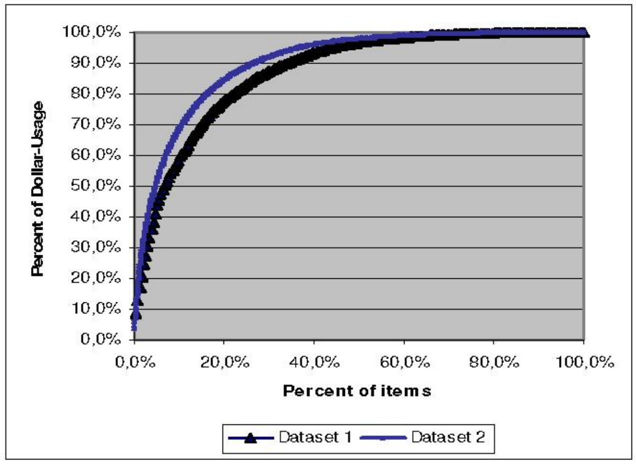
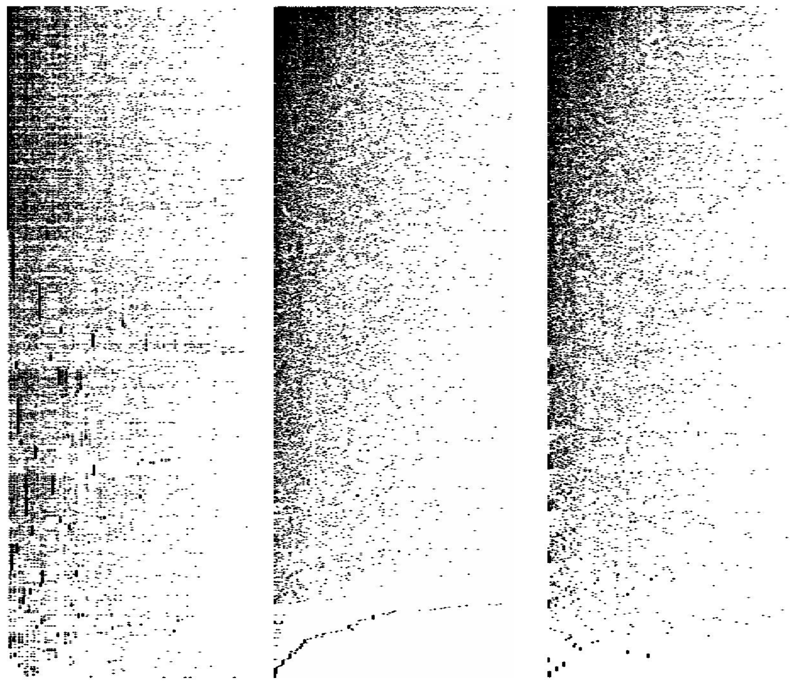


Figure 4. Distribution of dollar-usage values in datasets [13]

Results of applying the methods presented in this paper to real-world datasets (from [13]) are on Figure 5 and Figure 6. Such an approach gives us information about the product interdependencies, what is ignored by the classical ABC-diagram.

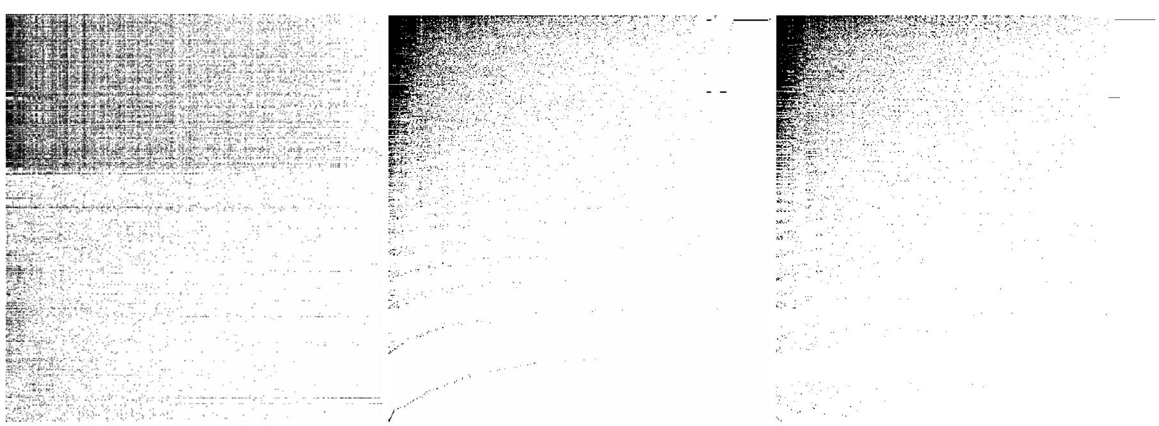
Initial datasets (matrices) are plotted directly to Figure 5(a) (Dataset 1) and Figure 6(a) (Dataset 2) - columns and rows denote stock keeping units and transactions, respectively. Corresponding dot is plotted if that product is bought, assembled or used in the specific transaction. Initial visualizations are similar to the results due to being pre-ordered according to the dollar-usage value. Differences between the initial visualization and the results demonstrate the impact of interdependencies between products.

Using the results of plain conformity calculation (Figure 5(b); Figure 6(b))) and complete processing of the algorithms (Figure 5(c); Figure 6(c)) we are able to refine the order of dollar-usage sorting and establish a two-mode typicality scale. With inventory transactions, clear co-behaving blocks seldom appear - typically the pattern of relations agglomerates in one corner, as 80/20 rule applies also to the customer behaviour. For example, rows on the top describe the most typical transactions in the system and the whole matrix visualization presents the transformation from common transaction to uncommon.



(a) (b) (c)

Figure 5. Dataset 1 [13] (SKU = 234, Transactions = 1465)



(a) (b) (c)

Figure 6. Dataset 2 [13] (SKU = 1601, Transactions = 1735)

V. CONCLUSIONS

In this paper we suggested new approach and algorithms for inventory management. A class of practical inventory analysis, visualization and enhanced demand association identification methods were presented and discussed. It is not suggested to replace the classical ABC-diagram (distributions of dollar-usage values), but to use the exploratory methods provided here to understand interrelations between products and customer behaviour - the transition of a typical transaction to untypical as well as item's typicality and co-occurrence in the transaction. Advantage of the approach presented here is that no parameter configuration (e.g. setting a threshold) is needed, which is otherwise done with test-and-trial strategy to establish domain-specific rules of thumb.

We can also observe from the experimental results in real-world scenarios that Pareto's principle (80/20) in inventories also holds to the underlying hidden natural order - majority of the relations and structure in the dataset is concentrated to a relatively small percentage of the objects.

Prototypes exist for all proposed extensions and are available for research and benchmarking purposes upon request.

REFERENCES

[1] H. F. Dickie, "ABC Inventory Analysis Shoots for Dollars Not Pennies," *Factory Management and Maintenance*, Vol. 109, No. 7, pp. 92-94, July 1951.

[2] B. E. Flores, D.C. Whybark, "Multiple Criteria ABC Analysis," *International Journal of Operations and Production Management*, Vol. 6, No. 3, pp. 38-46, 1986.

[3] B. E. Flores, D.C. Whybark, "Implementing multiple criteria ABC analysis," *Journal of Operations Management*, Vol. 7, No. 1/2, October 1987.

[4] G. W. Zimmerman, "The ABC's of Vilfredo Pareto," *Production and Inventory Management*, Vol. 16, No. 3, pp. 1-9, 1975.

[5] B. E. Flores, D. L. Olson, and D.C. Whybark, "Management of Multicriteria Inventory Classification," *Mathematical and Computer Modelling*, Vol. 16, No. 12, pp. 71-82, 1992.

[6] H. A. Güvenir, "A Genetic Algorithm for Multicriteria Inventory Classification," *Artificial Neural Nets and Genetic Algorithms, Proceedings of the International Conference*, Springer-Verlag, Wien, pp. 6-9, April 1995.

[7] H. A. Güvenir , E. Erel, "Multicriteria Inventory Classification using a Genetic Algorithm," *European Journal of Operational Research*, Vol. 105, No. 1, pp. 29-37, 1998.

[8] R. Q. Zhang, W. J. Hopp, C. Supatgiat, "Spreadsheet Implementable Inventory Control for a Distribution Center," *Journal of Heuristics*, Vol. 7, pp. 185-203, 2001.

[9] F. Y. Partovi, M. Anandarajan, "Classifying inventory using an artificial neural network approach," *Computers & Industrial Engineering*, Vol. 41, pp. 389-404, 2002.

[10] Q. Lei, J. Chen, and Q. Zhou, "Multiple Criteria Inventory Classification Based on Principal Components Analysis and Neural Network," *LNCS 3498*, Springer-Verlag, Berlin, pp. 1058-1063, 2005.

[11] R. Ramanathan, "ABC inventory classification with multiple-criteria using weighted linear optimization," *Computers & Operations Research*, Vol. 33, No. 3, pp. 695-700, March 2006.

[12] M. Cohen, R. Ernst, "Multi-item classification and generic inventory stock control policies," *Production and Inventory Management Journal*, Vol. 29, No. 3, pp. 6-8, 1988.

[13] I. Liiv, "Inventory classification enhancement with demand associations," *In Proceedings of the 2006 IEEE International Conference on Service Operations and Logistics, and Informatics*, pp.18-22, 2007.

[14] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases, " *In Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 207-216, 1993.

[15] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association in Large Databases, " *In Proceedings of 20th International Conference on Very Large Data Bases*, pp. 478-499, 1994.

[16] S. Brin, R. Motwani, and C. Silverstein, "Beyond Market Baskets: Generalizing Association Rules to Correlations," *In Proceedings ACM SIGMOD International Conference on Management of Data*, pp.265-276, 1997.

[17] R. T. Rust, V. A. Zeithaml, K. N. Lemon, *Driving Customer Equity*, New York: Free Press, 2000.

[18] L. Burbidge, "The New Approach to Production," *Production Engineer*, vol. 40, no. 12, pp. 769— 784, 1961.

[19] J. L. Burbidge, "Production flow analysis," *Production Engineer*, vol. 50, no. 4, pp. 139— 152, 1971.

[20] S. P. Mitrofanov, The Scientific Principles of Group Technology, 1966.

[21] J. McAuley, "Machine grouping for efficient production," *Production Engineer*, vol. 51, no. 2, pp. 53— 57, 1972.

[22] R. R. Sokal and P. H. A. Sneath, Principles of numerical taxonomy. W. H. Freeman, 1963.

[23] D. G. Kendall, "A Mathematical Approach to Seriation," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 269, no. 1193, pp. 125—134, 1970.

[24] U. Wemmerlöv and N. L. Hyer, "Research issues in cellular manufacturing," *International Journal of Production Research*, vol. 25, no. 3, pp. 413— 431, 1987.

[25] H. M. Chan and D. A. Milner, "Direct clustering algorithm for group formation in cellular manufacturing," *Journal of Manufacturing Systems*, vol. 1, iss. 1, pp. 65— 74, 1982.

[26] J. R. King and V. Nakornchai, "Machine-component group formation in group technology: Review and extension," *International Journal of Production Research*, vol. 20, pp. 117— 133, 1982.

[27] J. R. King, "Machine-component group formation in production flow analysis: an approach using a rank order clustering algorithm," *International Journal of Production Research*, vol. 18, iss. 2, pp. 213— 232, 1980.

[28] L. Vyhandu, "Rapid Data Analysis Methods," *Transactions of Tallinn Technical University*, No. 464, p.21-39, 1979.

[29] L. Vyhandu, "Some Methods to Order Objects and Variables in Data Systems," *Transactions of Tallinn Technical University*, No. 482, pp. 43-50, 1980.

[30] L. Vyhandu, "Fast Methods in Exploratory Data Analysis," *Transactions of Tallinn Technical University*, No. 705, pp. 3-13, 1989.

[31] I. Liiv, A. Vedeshin, E. Taks, "Visualization and Structure Analysis of Legislative Acts: A Case Study on the Law of Obligations," *In Proceedings of the 11th ACM/AAAI International Conference on Artificial Intelligence and Law (ICAIL'07)*, pp.189-190, 2007.

[32] I. Liiv, "Czekanowski-Bertin Learning Paradigm: A Discussion," *In Proceedings of the International Conference on Artificial Intelligence (ICAI'07)*, pp. 90-95, 2007.