INFORMACIJOS SISTEMOS IR MODELIAVIMAS

Making of a Duty Schedule Using Action Rules*

Darius Jurkevičius

Vilniaus Gedimino technikos universiteto doktorantas
Vilnius Gediminas Technical University, Doctoral student
Saulėtekio al. 11, LT-10223, Vilnius
El.paštas: d.jurkevicius@isl.vgtu.lt

Olegas Vasilecas

Vilniaus Gedimino technikos universiteto profesorius, daktaras
Vilnius Gediminas Technical University, Professor, doctor
Saulėtekio al. 11, LT-10223, Vilnius
Tel. (+370 5) 274 48 60
El.paštas: Olegas.Vasilecas@fm.vgtu.lt

The system for duty schedule generation and optimization solves a difficult, important and practical problem. This system allows the user not only to see the initial and optimized duty schedule in a simple and understandable form, but also to create special rules and criteria, and allows inputting their business. The system allows collecting participant's information about their occupation during a month. The system using the action rules and formal concept analysis will automatically generate a duty schedule.

1. Introduction

There are many tools used for schedule creation. But these tools are pointed to create duty schedules irrespective of the additional occupation of the staff. These tools are used when the work is organized on the on-duty basis. Some institutions organize additional work involving being on-duty, for example, army (The General Jonas Zemaitis Military Academy of Lithuanian), where training is running together with arming. Sometimes monthly on-duty schedules are needed; in that case, the monthly additional occupation of the staff must be taken into account. This data must be picked up and analyzed, and afterwards schedules are created.

One of the ways to create a schedule taking into account these limitations is using the rules. The rules can be used to describe the status of employees. The Microsoft Excel was used for creating a duty schedule. It was a great problem for a man who has been creating a duty schedule because he could not exactly evaluate all parameters.

Other parts of this paper consist of these sections:

- Formal concept analysis and its usage are described in the first section.
- Multiple (distributional) formal context is described in the second section.
- The method of creating a duty schedule is described in the third section.
- The experiment is described in the fourth section.
- Conclusions are presented in the last section.

* The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "Business Rules Solutions for Information Systems Development (VeTIS)" Reg. No. B-07042.

2. Understanding of the formal concept analysis

One of the ways of transforming the available data in a hierarchic aspect is a formal concept analysis. Dau (2004) has noticed that scientists could not lean on them as on arguments in making plots. To discriminate formally the mathematical structure from its scheme presentment, the work environment was created in which diagrams could be used to make formal substantiations. We define here some terms used in this article:

Concept can be defined as:

– an abstract or general idea inferred or derived from specific instances (Wordnet, 2009);

– a concept is an abstract idea or a mental symbol, typically associated with a corresponding representation in language or symbology, that denotes all of the objects in a given category or class of entities, interactions, phenomena, or relationships between them (Encyclopedia Britannica, 2009);

– it has an intention (deep definition), extension (set of objects or exemplars) (Martin, 1994; Odell, 1994)

– definition of the type of objects or events. A concept has an intensional definition (a generalization that states membership criteria), and an extension (the set of its instances) (Mayers, 2004; Maulsby, 2004)

Formal Concept Analysis (FCA) (Wille, 1982) method is:

– mathematization of the philosophical understanding of concept;

– a human-centred method to structure and analyze data;

– a method to visualize data and their inherent structures, implications and dependencies.

FCA is based on the philosophical understanding that a concept can be described by its extension – that is, all the objects that belong to the concept and its intension which are all the attributes that the objects have in common (Tilley, 2003);

Formal context – the mathematical structure used to formally describe these tables of crosses (or briefly a context) (Wolf, 1993).

FCA is a method used in data analysis, knowledge imaging, and information control. Rudalf Wille proposed FCA in 1981 (Wille, 1982), and it is successfully developed nowadays. For 10 years FCA has been researched by small groups of scientists and Rudalf Wille's students in Germany. FCA was not known worldwide because the largest part of publications was presented at mathematicians' conferences. After getting the sponsorship, some projects have been implemented in this area. Most of them were knowledge research projects used in work. This system was known only in Germany. During the last 10 years FCA has become the research object of international scientists' community. FCA was used in linguistics, psychology, as well as in projecting the software, in the areas of artificial intelligence, and information search.

Some of the structures of FCA appeared to be fundamental to information representation and were independently discovered by different researchers. For example, Godin (Godin, 1989; Gecsei, 1989; Pichet, 1989) use of concept lattices (which they call "Galois lattices") in information retrieval.

Here we introduce a definition of the formal concept analysis (Ganter, 1999; Wille, 1999).

For example, G is the set of objects that we are able to identify in some domain (e.g. if, when, than). Let M be the set of attributes. We identify index I as a binary relationship between two sets G and M, i.e., $I \subseteq G x M$. A triple (G,M,I) is called a formal context. For $A \subseteq G$, we define:

$$A' := \left\{ m \in M \;\middle|\; (g,m) \in I \;\; for\ all\ \; g \in A \right\} \quad (1)$$

and dually, for $B \subseteq M$

$$B' := \left\{ g \in G \;\middle|\; (g,m) \in I \;\; for\ all\ \; m \in B \right\} \quad (2)$$

A formal concept of the formal context (G,M,I) (Fig 1.) is defined as a pair (A,B) with $A \subseteq G$, $B \subseteq M$, $A' \subseteq B$ and $B' \subseteq A$. Sets A and B are called extend and intend of the formal concept. The set of all formal concepts of the context (G,M,I) is called a concept lattice (Fig. 1) of the context (G,M,I).

	female	juvenile	adult	male
girl	x	x		
woman	x		x	
boy		x		x
man			x	x

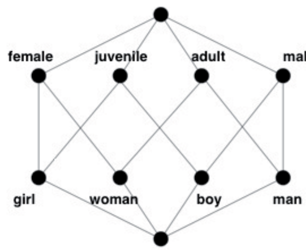

Figure 1. ***An example of the formal context and a concept lattice (Galois lattice)***

3. The method overview

Now we will describe the proposed method how to make a duty schedule using the formal concept analysis for data analysis and rules.

The main idea of creating a duty schedule is to find the greatest conflict objects (people) and attributes (number of days). The intersection of these is the greatest conflict that we had found. Then the number of the found conflicts will reduce.

Algorithm of the schedule's (Fig. 2):

1. Make the monthly day-off list depending on the number of people being on duty that day (Fig. 4a);
2. Make the monthly list workdays depending on the number of people being on duty that day (Fig. 4b);
3. Organize a list depending on the number of persons being on duty this year;
4. Select the day;
5. If there are free days in the list, then go to 8;
6. If there are no free days in the list, then go to 17;
7. Make the list of objects (people) depending on the number of days when a person cannot be on duty (Fig. 4);
8. If the list of objects is empty, then go to 14;
9. If the list of objects is not empty, then go to 10;
10. Organize a list depending on the number of weights of business;
11. Select the object (person) from the list and check it (Fig. 3 and 5);
12. If this can be labelled, then insert and go to point 4;
13. If this cannot be labelled, then go to point 11;
14. Make a list of all the objects;
15. Organize a list depending on the number of weights of business;
16. Select and insert the first object from the list and go to 4;
17. The end.

The checking process of the rules is shown in the dotted area of Figure 2. This process is explained in Figure 3.

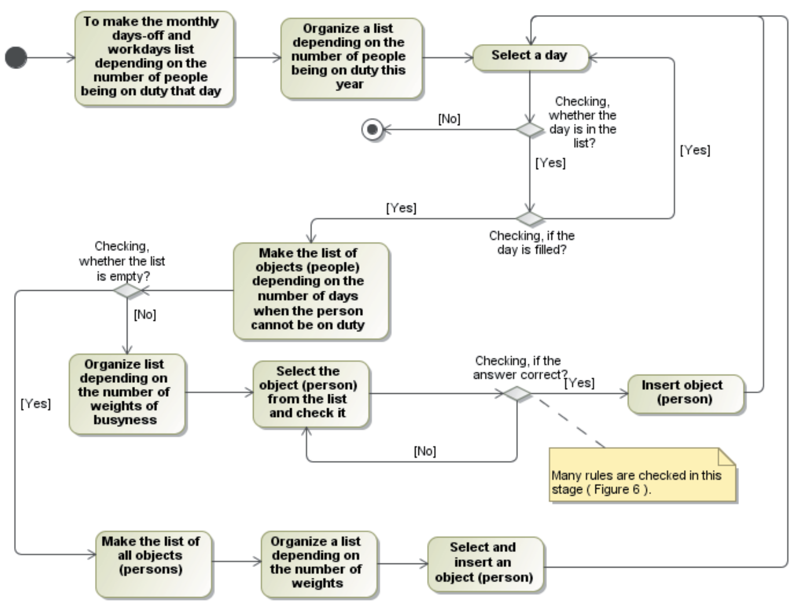

Figure 2. ***Algorithm of a duty schedule***

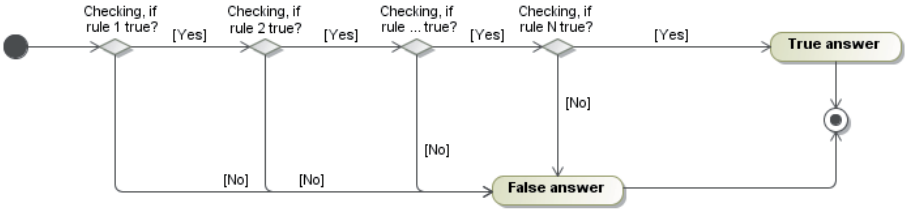

Figure 3. ***Explanation how to check the objects using the rules***

For example, after the data analysis (Fig. 4a,b) we have got a sequence of attributes (days): 2, 1, 8, 9 (day-off) and 6, 5, 4, 3, 7 (work-day). Another data analysis is shown in Figure 8, where k is attribute (day) numbers. For example, for day 4 we got a sequence of objects (persons): 5, 6, 7, 9, 13, 14. Then these objects are analyzed and checked. This stage is showed in Figure 8. Object 5 wins because its final weight was lower (0) than the other. Then the object inserted in the data base wins.

Figures 3 and 5 show an example how to check objects. One stage is check of the rules, and the other stage is check of weights. Sources of the rules are a personal section, study schedule, etc. There is a reason to use the rule machine (for holding rules) because some kinds of rules are changing quite quickly. For example, the rules in the natural language:

- Person (John) cannot be on duty from January to February;
- People holding certain appointments can be on-duty just during a day-off;

When a user describes rules, then these rules can be ambiguous and without definition. We can remove indefiniteness and ambiguity, if each rule is resolved into an elementary or atomic rule. The atomic business rule is declaratively written, using a natural language, easily understood by businessmen, and it is not ambiguous. Information system designers write atomic rules using a formal language. During the stage of rule transformation, cross-purposes can occur because users have their own language, and system creators have their own. Mistakes occurring during the stage of the rule transformation process can be removed if a user himself writes the declarative rule by the template proposed in the natural language.

When inputting the rule in the declarative form, it is suggested the template, or this rule can be written using a semi-structured natural language.

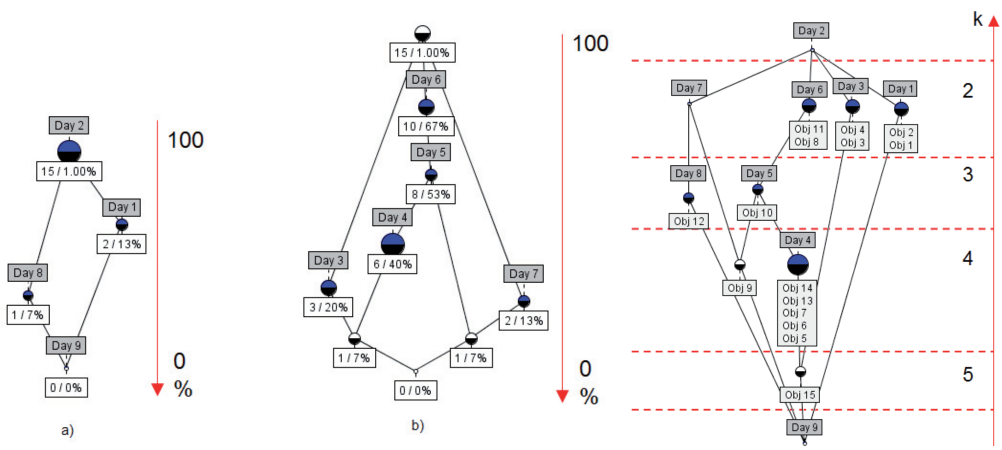

Figure 4. ***Generation of a lattice of concepts and their organization according to the attributes. a) according to the day-off; b) according to work-day. And generation of concept lattice and their organization according to objects (the picture on the right side)***

All objects belong to day 4

Objects:	5	6	7	9	13	14
Weights:	**0**	1	-	5	-	-
Result of checking the rules:	**1**	1	0	1	0	0

Figure 5. ***Example of object analysis (object 5 is selected as best)***

Using the template input proposed by the system, the initial component of the rule (ex. If) is suggested (Jurkevičius, 2009; Vasilecas, 2009). Then other terms that are kept in the formal context are suggested. It is a step-by-step constructed rule. A mistake can be avoided using the first method, when the templates are proposed, and the rule can be immediately transformed into the formal form. This method is designed mainly for the businessmen. Hereunder we show that different rules have a generic form. Rules can be defined by declarative IF-THEN form with the priority, start date, finish date, for example:

The first rule:

- If a person is John and the month is January or February, then reject;

The second rule:

- If a person is studying and seeking the master degree and now is a work-day, then reject;

At first sight, these rules do not have much in common. After analyzing the rules, it has been recognized what condition and action of the rule can be composed of some components.

If we write this rule in the most usual way, it would be:

If condition, then action.

There are not so many ways of formulating a condition and action.

4. Experiment

A prototype has been created to make duty schedules and manage them. This tool, a web application, is called the duty schedules manager (DSM) (Figure 6). That tool is used in General Jonas Zemaitis Military Academy of Lithuanian.

DSM has a complete feature set to enable you to create your own schedule.

The main features of DSM are:

- Generating the schedule automatically;
- Editing the schedule and choosing another alternative;
- Possibility to create the same duty schedules;
- User can get his/her own data about occupation;
- Users have roles in the system.

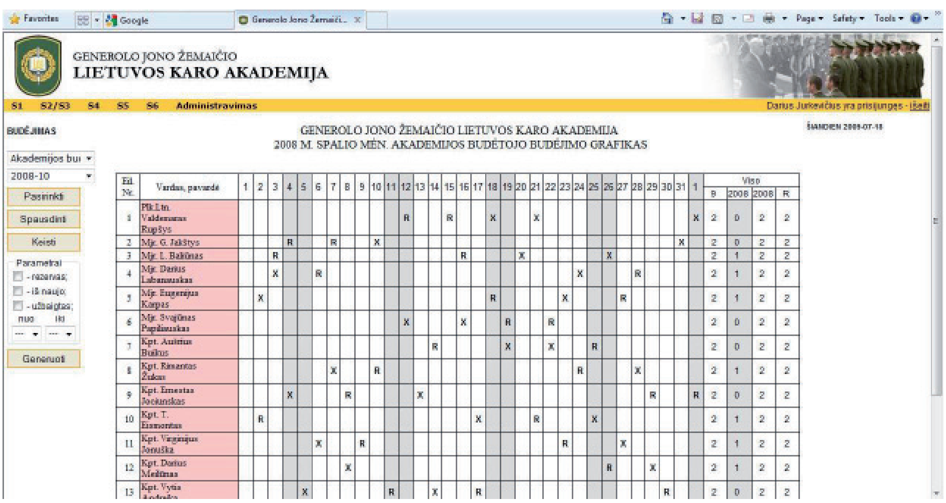

Figure 6. ***The application for creating a duty schedule***

5. Conclusions and future plans

Using the proposed method, the user is able to get his/her data about occupation. The schedule is generated more precisely and quickly than the paper schedule.

Our future plans are to explore how to make the rule management easier.

REFERENCES

DAU, F. (2004). Types and Tokens for Logic with Diagrams. In: K. E. Wolff, H. Pfeiffer, & H. Delugach (Eds.). *Conceptual Structures at Work: 12th International Conference on Conceptual Structures.* Berlin: Springer, p. 62–93.

JURKEVIČIUS, D.; VASILECAS, O. (2008). Rules transformation using formal concept approach. In: Papadopoulos, G. A., Wojtkowski, W., Wojtkowski, W. G., Wrycza, S., & Zupancic, J. (eds). *Proc. of the 17th International Conference on Information Systems Development* (ISD2008) – Information Systems Development: Towards a Service Provision Society, New York: Springer-Verlag.

MAYERS, A.; MAULSBY, D. (2004). Glossary. [Overviewed on 2007-12-11]. http://acypher.com/wwid/BackMatter/Glossary.html.

GANTER, B.; WILLE, R. (1999). *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin–Heidelberg.

GODIN, R.; GECSEI, J.; PICHET, C. (1989). Design of browsing interface for information retrieval. In N. J. Belkin, & C. J. van Rijsbergen (Eds.). *Proc. SIGIR '89*, p. 32–39.

MARTIN, J.; ODELL, J. (1994). *Object-Oriented Methods: A Foundation*. Prentice-Hall, p. 52.

WILLE, R. (1982). Restructuring Lattice Theory: an approach based on hierarchies of concept. In: *Ordered sets* / editor I. Rival.— Reidel, Dordrecht-Boston, p. 445–470.

TILLEY, T. (2003). *Formal Concept Analysis Application to requirements engineering and design* [overviewed on 2007-11-15]. http://www.int.gu.edu.au/~inttille/publications/ tilley04formal.pdf.

WOLF, K. (1993). A first course in formal concept analysis. In: Faulbaum, F. (ed.) *SoftStat'93 Advances in Statistical Software* 4, p. 429–438.

WORDNET (2009) a lexical database for the English language [overviewed on 2008-01-08]. http://wordnet.princeton.edu/ perl/webwn?s =concept.

ENCYCLOPEDIA BRITANNICA (2009), Concept. The Classic Encyclopedia based on 1911 Edition of the Encyclopedia Britannica [overviewed on 2009-07-18]. http://www.1911encyclopedia.org/Concept.

BUDĖJIMO TVARKARAŠČIŲ SUDARYMAS NAUDOJANT FORMALIŲ KONCEPTŲ ANALIZĘ IR VEIKLOS TAISYKLES

Darius Jurkevičius, Olegas Vasilecas

S a n t r a u k a

Budėjimo tvarkaraščiui sudaryti yra sukurta daug įrankių. Tačiau šie įrankiai yra taikomi neatsižvelgiant į papildomą darbuotojų užimtumą. Kai kuriose organizacijose papildomai darbas organizuojamas įtraukiant budėjimus. Vienas tokių pavyzdžių yra kariuomenė (Generolo Jono Žemaičio Lietuvos karo akademija), kur mokymas organizuojamas kartu su karo tarnyba.

Dažnai tenka sudaryti keletą budėjimo tvarkaraščių vienam mėnesiui. Sudarant tvarkaraščius, reikia atsižvelgti į darbuotojų užimtumą. Šiuos duomenis reikia surinkti, išanalizuoti ir jais remiantis sudaryti tvarkaraščius. Vienas iš būdų sudaryti tvarkaraščius atsižvelgiant į pirmiau minėtus ribojimus yra taisyklių naudojimas. Taisyklėmis yra aprašomos darbuotojų būsenos laiko atžvilgiu. Straipsnyje pateiktas metodas, kaip sudaryti tvarkaraščius pasitelkiant veiklos taisykles ir formalių konceptų analizę.