Triadic Formal Concept Analysis within Multi Agent Systems ⋆

Petr Gajdoš, Pavel Děrgel

Department of Computer Science,
VŠB - Technical University of Ostrava,
tř. 17. listopadu 15, 708 33 Ostrava-Poruba
Czech Republic
Petr.Gajdos@vsb.cz
Pavel.Dergel@vsb.cz

Abstract. This article deals with a usage of triadic Formal Concept Analysis within Multi Agent Systems (MAS). MAS are hot topic of a current research, although their theoretical background is quite old. They become more popular in many branches, because they provide solutions of problems that are too extensive and time-consuming. They provide a connection and a cooperation between several systems. MAS data can be analyzed by FCA. On the other hand, features of MAS enable us to distribute the computation of concept lattices over several hardware platforms. A particular usage of FCA and MAS is mutual as described in this article.

Keywords: Multi Agent System, Formal Concept Analysis, Triadic Concept Lattice

1 Introduction

Multi agent systems (MAS) become more popular now because of their features that allow to solve problems related with the size of systems, number of users etc. We tried to use the Triadic Formal Concept Analysis to make a system preview and analyze the data within MAS. However, new problems arisen during the FCA implementation, e.g. the problems connected with the size of input data set. We decided to use MAS itself to solve such difficulties. Finally, a mutual usage of FCA and MAS gave rise to a new application. This article will shortly describe the usage of FCA for a MAS data analyzing as well as the usage of MAS for a FCA distributive computation.

For the necessary definitions of triadic FCA we refer to [9] [3] [2] [6]. Multi Agent Systems and main terms from this area will be defined. The concrete usage of FCA will be shown on a small illustrative example. The FCA integration within MAS will be described in a separated section.

⋆ This research has been supported by the program "Information Society" of the Czech Academy of Sciences, project No. 1ET101940420 "Logic and Artificial Intelligence for multi-agent systems".

2 Multi Agent Systems (MAS)

Before presenting general definition of a Multi Agent System, we first introduce basic units of such systems. An *agent* represents an essential part of all Multi agent Systems. There are several different definitions of an agent, we cite one of them:

Definition 1. *An* ***agent*** *is a programme process that implements own autonomy and communication capability. Agents communicate by the usage of a communication language* ACL *(Agent Communication Language). An agent has to have at least one owner, i.e., one organization or user, and has to have a defined identity AID (Agent Identifier) that is unique. This identity helps the user to exactly identify an agent in the whole system. [4]*

As mentioned above, there are many other definitions of an agent. However, we can find some common points:

- An agent is situated in a specific environment and it is able to accept or respond to the outer stimuli.
- An agent is able to work autonomously.
- It is able to move, clone itself or remove itself from the environment.
- It is able to communicate with the others agents.

So, an agent is a computer programme that is situated in some environment and it is able to work autonomously. However, *autonomy* is very difficult notion. In the context of MAS it means that an agent is able to work independently of the others, and if a particular agent malfunctions the whole system does not collapse.

Design of a multi agent system which is exclusively reactive or proactive is an easy task. However, when attempting at a combination of both and their proper balance, we meet problems. Agents could achieve their objectives systematically. We do not want the agents to blindly perform a sequence of procedures or functions. We need that the agents react dynamically and do not try to do something that is already impossible. The last aspect of agents' intelligence consists in the cooperation as was mentioned in the work of Zambonelli [10]. It is not the data exchange only. It is a common term that includes communication, negotiation and cooperation. The whole problem of agents' cooperation is the most challenging task and it has still not been solved in a satisfactory way.

2.1 MAS - specification

The skills of intelligent agents are limited by their knowledge, computer resources and surrounding behaviour; agents are resource-bounded. A single agent is usually not able to perform a complex real-world processes. We need a couple of agents to simulate such processes. Groups of co-operative agents make up a system called *Multi Agent System*.

MAS is a hot topic of current research. Because these agents are apt for applying in many areas. Here is a list of some advantages of MAS:

- MAS provide the solution of problems that are too extensive and time-consuming for classic realized system.
- There is a possibility for connection and cooperation between several systems.
- They work with distributed information, e.g. sensor monitoring.
- The agents can be mobile within a system. They are able interrupt their activity, move into another place within the current system and then continue the work in a new locality.

In the following text, the term *mobile agent* means an such agent that can change its position in a real word, i.e., it can change its real coordinates. An agent can never have a complete knowledge of the whole environment. The model of such environment is a crucial problem and it plays an important role within all the experiments with a Multi Agent System.

3 Triadic FCA within MAS

An application usage of FCA will be demonstrated on particular Multi Agent System. The following text describes possible tasks that could be solved by the MAS.

3.1 Problem specification

For the sake of simplicity, let us consider *accessible*, *deterministic*, *static* and *discrete* environment of MAS. Let define the main part of such MAS:

- **O**: the set of objects, namely static elements of the environment, e.g. roads and crossroads.
- **A**: the set of agents. We consider *mobile* agents such as cars that can move along the environment. More precisely they move along the objects, such as roads.
- **R**: the set of relations between agents and objects. It represents a possible movement of some agent along some object. We can find many restrictions of such movements. For example, agents can have features which impose limitations on their movement (a tractor cannot move along a motorway because of its speed, or the trucks cannot be on a village way from 16:00 PM to 6:00 AM next day).

Example 1. Triadic lattice in MAS Consider the 3-ary relation $R3 \subseteq WHERE \times WHAT \times WHO$ for the usage of triadic FCA in the above defined MAS. The concrete usage will be shown on a small instance of an imaginary city that will simulate the reality. The example is focused on the traffic simulation.

The whole geographical area is covered by several roads and crossroads. Each road type is connected with traffic restrictions, e.g. trucks can not move along the lanes. Such a system of roads represents a static structure of MAS. Hence in the above defined MAS, each road or crossroad is represented by an object

$o \in O$. The inhabitants of the city make up the dynamic part of the system. Each inhabitant is an agent $a \in A$ of MAS. Then we have following context $\mathbb{K} := (G, M, B, Y)$, where G is a set of places, M is a set of traffic violations, B is a set of agents and Y is a ternary relations that represents the above defined kind of the relation $WHERE \times WHAT \times WHO$. $(g, m, b) \in Y$ if and only if an agent $b \in B$ has committed a traffic violation $m \in M$ on a place $g \in G$. Then the triadic concepts can be computed.

After having constructed the three dimensional incidence matrix, a concept list and a triadic lattice are computed. The process of such a computation as well as its integration into MAS will be described in the next section. However, the resulting concept list gives rise to several questions. Particular concepts can be analyzed. We look for common features of agents' behaviour, that lead to committing traffic violations.

4 FCA integration within MAS

Particular implementation of FCA within MAS depends on many aspects, however, the main idea consists in a direct integration of FCA algorithms with MAS framework. An alternative approach consists in the representation of FCA as an agent of such a framework; this is the way we have chosen because it is in accordance with the FIPA specification [5].

4.1 MAS framework

The MAS framework is a kernel of this kind of systems. It has to be based on standards that make it widely used without any platform dependency. FIPA [5] (The Foundation for Intelligent Physical Agents) is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies. For our purpose, JADE (Java Agent DEvelopment Framework) has been chosen. It is a software Framework fully implemented in Java language, however, has been rewritten into Microsoft .NET J# and C#, respectively. JADE is a middle-ware that simplifies the development of applications. Several companies are already using it for very different application sectors including a supply chain management, rescue management, fleet management, auctions, tourism, etc.

4.2 Simplified scheme of FCA integration

Now, we will continue in the example 1. In the following scheme (see the figure 1) we decided to use another representation of static elements of MAS environment. The roads and crossroads (the static elements earlier represented by objects) will now be represented by agents. The reason for this is very simple. We need to receive information on an accident and somebody has to send it. Objects do not do that. According to the definition, they form the passive part of MAS and only agents can send a message. In the real world there are cameras, sensors, etc. which can be used to obtain a view of the current situation on the roads or crossroads. Of course, functions of such facilities can be simulated, however, it does not come within the ambit of our interest.

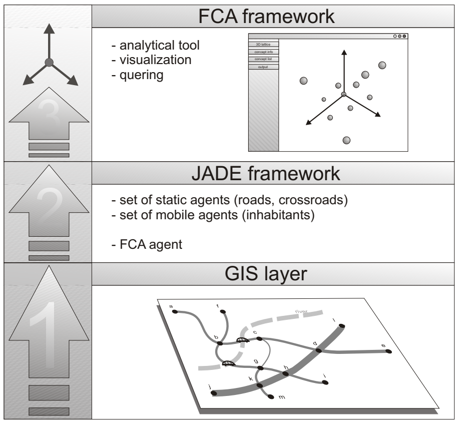


Fig. 1. The basic scheme of FCA integration

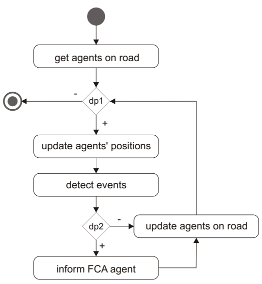
The following points summarize the basic description of each level (see the figure 1):

1. At the low level, GIS (Geographical Information System) and its methods are used as a subsidiary tool. From a GIS database, the set of roads, crossroads and their connections is obtained.
2. The second level is represented by some MAS framework. As mentioned above, JADE has been chosen. First, JADE platform is started, which initializes *agent containers* (see more details in [1]). Second, one agent for each road and crossroad is created. Third, some "inhabitants"-agent whose activities are to be executed are also created. Finally, the "FCA"-agent is started. FIPA [5] defined basic elements of MAS as well as the structure of agents.
3. The top level consists of the FCA framework. It is a set of tools that comprises the 3D lattice visualization, FCA algorithms for biadic and triadic concept analysis including improved algorithms with a new storage system, analytical tools, etc. It allows the users to see the current situation of MAS in a tri-lattice.

4.3 Agents' interaction and communication

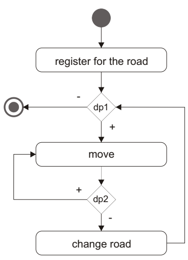
When talking about agents integration and communication we will keep the manner of the FIPA [5] specification. More precisely, we will talk about the way FCA agents receive information on an accident, failure, etc.

Each agent has defined at least one *behaviour* before it is initialized. JADE makes it possible to define several types of behaviour but we need only one of them. Our agents will have only the so-called *Cyclic behaviour* defined. This kind of behaviour consists of a sequence of *activities* that are repeated. The figures 2, 3, 4 show simple behaviour of three types of agents (static, mobile, FCA) that we used. Related descriptions are on the right from the schemes represented by *activity diagrams* [8].



At the beginning, a static agent "A" has to get all information on all agents that move along the agent "A". In case there is no agent (see dp1) the behaviour is finished and the agent is suspended until a new mobile agent sends a registration message. Otherwise the positions of all agents moving along our road/crossroad are updated. This behaviour is responsible for the real word simulation. It means that the action *detect events* recognises all possible events (e.g. car accident, speeding, etc.). If an event occurs (see dp2), the information on it is sent to the FCA agent. Finally, the static agent looks for mobile agents on road/crossroad again and updates its data.

Fig. 2. The behaviour scheme of static agents (e.g. roads, crossroads).

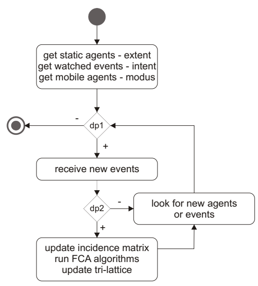


The behaviour starts with the registration activity. Each agent has to have a position in our system. If the registration fails (see dp1), the behaviour is finished and the agent is *killed* (the term from FIPA specification [5]). Otherwise the agent can move along the registered static structure (e.g. road, crossroad). Agent can make the next step of its movement if and only if it does not reach the end of the current road/crossroad. Otherwise it has to change its location (see dp2).

Fig. 3. The behaviour scheme of mobile agents (e.g. inhabitants).

4.4 Problems connected with the usage of triadic approach and their solutions

There are also some problems connected with usage of triadic FCA. Of course, it is primary the size of 3D cross table that represents the context and the size



The initialization phase of this behaviour consists in creation of an empty context $\mathbb{K} := (G, M, B, Y)$, where G is a set of static agents, M is a set of watched events, B is a set of mobile agents and Y is the relation $G \times M \times B$. If the initialization fails, the behaviour is finished and agent is killed (see dp1). Otherwise the FCA agent tries to receive a new piece of information on events. If an event occurs in our system (see dp2) the context, as well as list of all triadic concepts and concept lattice are updated. After that, FCA agent looks for new agents or events which could be created in the whole system.

Fig. 4. The behaviour scheme of FCA agent.

of a list of concepts and tri-lattice, respectively. However, usage of MAS drove us to solve these problems in another way. It will be described now.

- **Agent cloning** is one of the basic features of all MASs implemented according to FIPA standards. This feature allows us to create more than one FCA agent with the same functionality. Each of them can process information from different, disjoint areas. In addition, each agent can be located on different hardware platforms, which is made possible by using JADE. Also, they do not have to share processor time, memory or storage space. In case of need, a new "Chief FCA agent" can be created that merges partial contexts and lattices, respectively. This can be done by scalable *divide-and-conquer* methods made by Valtchev et al. [7]. The figure 5 illustrates such distribution.
- **Message filtering** is an other way of reducing the size of a context. FIPA defines the following structure for all messages within MAS, e.g. *sender*, *receiver*, *content*, *language*, *ontology*, *protocol*, etc.

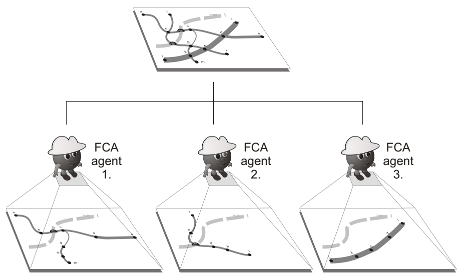


Fig. 5. The scheme of FCA distribution using MAS technology.

5 Conclusion and future work

The usage of triadic FCA within MAS has been illustrated by traffic example. Of course, there are other applications of FCA within MAS. The above mentioned ternary relation $R3 \subseteq WHERE \times WHAT \times WHO$ can be interpreted in another way. Regardless of the way the relation is interpreted within a particular MAS, the merits of using FCA and the principles of its functionality are significant.

In this chapter, we described particular FCA implementation within MAS. Such implementation and usage help us to design and observe the agents and their behaviour. The size of triadic lattice is itself informative. In case of a small triadic lattice there are a few traffic violations on the roads, which suggest that the agents are well designed. Their behaviour (inner processes) do not allow to violate the laws. Moreover, we may investigate significant concepts some interesting properties.

The mutual interpretation of FCA and MAS was also described. The FCA can be used as an analytical tool within MAS, whereas the features of MAS allow us to distribute and simplify lattice computation.

References

1. F. Bellifemine. Java agent development framework documentation. http://jade.tilab.com/, 2005.
2. K. Biedermann. How triadic diagrams represent conceptual structures. In *Proceedings of the 5th International Conference on Conceptual Structures*, volume 1257 of *LNAI*, pages 304–317, Berlin, 1997. Springer.
3. K. Biedermann. Powerset trilattices. In *Proceedings of the 6th International Conference on Conceptual Structures: Theory, Tools and Applications (ICCS-98)*, volume 1453 of *LNAI*, pages 209–224, Berlin, 1998. Springer.
4. F. T. F. for Intelligent Physical Agents. http://www.supercomp.org/sc95/proceedings/473_mber/sc95.htm, 2005.
5. J. Odell. The foundation for intelligent physical agents. http://www.fipa.org/, 2006.
6. L. Schoolmann. Triadic concept graphs and their conceptual contents. In B. Ganter and R. Godin, editors, *ICFCA*, volume 3403 of *Lecture Notes in Computer Science*, pages 285–298. Springer, 2005.
7. P. Valtchev, R. Missaoui, and P. Lebrun. A partition-based approach towards constructing galois (concept) lattices. In *Discrete Mathematics*, volume 3, page 801829, 2002.
8. I. Vondrák. *Methods of Business Modeling*, volume I. VŠB-TUO, Ostrava, 2004.
9. R. Wille. The basic theorem of triadic concept analysis. *Order 12*, pages 149–158, 1995.
10. F. Zambonelli. Developing multi-agent systems: The gaia methodology, 2003.