Active Graph Rewriting Rules for Modeling Multi-Agent Organizational Dynamics

Markus Schatten
University of Zagreb
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia
E-mail: markus.schatten@foi.hr

Abstract
The field of multi agen systems' (MAS) organizational design deals with providing methods for building organizational architecture of MAS organizations. Herein, an active graph grammar (AGG) formalism, inspired by current research in graph and active database theory is introduced and applied to modeling MAS organizational structure. By defining organizational units in a recursive way, it is shown that labeled graphs and hypergraphs can be used to model various levels of organizational structure. The newly developed method is graphical, event-driven and applicable in a distributed MAS environment.

Keywords: multi agent systems, organizational dynamics, active graph grammar, organizational architecture, graphical method

JEL classification: C0, C61, C88

1. Introduction
Multi agent systems (MAS) are a well established abstraction for modeling distributed computing systems. MAS consist of a number of interacting agents, which act according to the goals and motives of their users. In order to achieve their goals, agents need the abilities of cooperation, negotiation as well as coordination. An important issue in MAS design is the development of agents, which are able to interact with other agents, in order to achieve their goals. This is the field of MAS organizational design.

In the following we will focus on this problem by using a well established formalism, namely graph grammars (GGs) which allow us to capture and program structural regularities (Nagl, 1979). The idea of applying GGs to MAS is not a new one. For example Nagendra Prasad et al. (1996) present a GG based task structure specification language for the TÆMS agent models. They went on and extended GGs with stochastic and attributed productions. Nagendra Prasad and Lesser (1999) later on used this tool to model the topological relationships occurring in task structures, as did Lesser et al. (1999) to implement a MAS for managing an intelligent environment as part of the intelligent home project (IHome). Giese et al. (2003) employ story diagrams introduced by Fischer et al. (2000)[1], to model safety-critical MAS macro- and

[1] Story patterns in fact illustrate graph grammar formulae.

micro-architectures in UML. Later Becker et al. (2006) extend the approach to develop a verification technique for arbitrarily large multi-agent systems in mechatronics. Smith et al. (2009) use embedded graph grammars to deploy and coordinate robots (agents) in various (physical) formations. They introduce an agreement protocol for agents to agree mutually before applying a production rule.

While these studies provide important particular insights into the matter, herein we will introduce a more holistic approach. Firstly, we shall define MAS organizations in terms of organizational architecture[2]. Thus MAS are not only collections of agents structured in a certain way, but an agent organization consisting of organizational structure, organizational culture, strategy, processes and individual agents (human resources in humane organizations). These five perspectives represent important and different views of the same (agent or humane) organization:

Organizational structure defines the decision and information flows of an organization.

Organizational culture defines important intangible aspects of an organization including knowledge, social norms, reward systems etc.

Strategy defines the overall objectives of an organization as well as tools on how to measure success.

Processes define the activities and procedures of an organization.

Individual agents define the most important asset of any organization - the individuals actually performing the work.

All of these perspectives are subject to changes due to changes in the environment of the organization. Organizational change is probably one of the most important aspect of successful organizations. In the following we will restrain ourselves on organizational dynamics of organizational structure, but with smaller modifications most of the approach outlined herein can be applied to all the other perspectives.

The previous outlined studies, deal mostly with individual agents as part of an agent organization, whereby each agent takes a certain role. Herein we will use the fractal organization principle (Warnecke, 1992) which allows us to define organizational units recursively.

Definition 1 *An organizational unit is defined as follows:*

- *Any agent is an organizational unit.*
- *If $O = \{o_1, o_2, ..., o_3\}$ is a set of organizational units which collaborate with a common objective, then O is an organizational unit.*

This definition is very subtle since it allows us to deal with agents, groups/teams of agents, organizations of agents, networks of organizations of agents as well as virtual organizations of agents[3] in the same way. Note that the term objective here is arbitrary and could easily be replaced with function, goal, mission, unit name etc.

In the following we will make use of this definition to apply a formalism for modeling organizational dynamics in MAS. The rest of this article is organized as follows: in section 2 we formalize active graph rewriting rules where we will extend the previous GG approaches with an active component. Section 3 gives a modeling example of the previously introduced formalism. In section 4 we draw our final conclusions and give guidelines for future research.

[2]Please refer to (Žugaj and Schatten, 2005) for an in-depth discussion on organizational architecture.

[3]Please refer to (Barnatt, 1995) for our understanding of virtual organization as an overlay structure.

2. Active Graph Rewriting Rules

Inspired with active database theory, we will introduce active graph rewriting rules (AGRR) over labeled graphs. We will define labeled graphs in an (unusual) object-oriented way.

Definition 2 *A label is an ordered pair $a : v$ in which a is an attribute and v is a value.*

Definition 3 *Let N be a set of nodes, $E \subseteq N \times N$ a set of edges, $L = L^N \cup L^E$ a set of labels (whereby L^N is a set of node labels and L^E a set of edge labels). Let furthermore $\nu \subseteq N \times L^N$ and $\varepsilon \subseteq E \times L^E$ be two corresponding relations which map nodes and edges to their labels respectively. We denote the tuple $(N, E, L, \nu, \varepsilon)$ as a labeled graph.*

To further foster object-orientation we will extend node labels with methods.

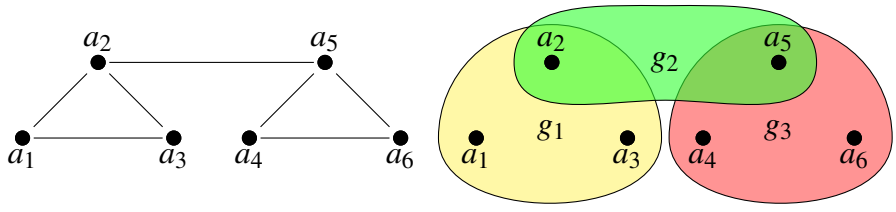
Definition 4 *A method label is the ordered pair $m/a : v$ in which m is a function of arity a and v is a return value. Usually the first argument to the function is a reference to the current node.*

In the following we will model MAS using labeled graphs, whereby organizational units are denoted with nodes, and their interaction with edges. In order to model MAS organizations, we will introduce two special labels: (1) c being an edge label denoting that two organizational units (nodes) which are connected through such a label collaborate, and (2) g being a node label which denotes the goal of the node. Note that g could easily be a method function for various situations in which an organizational unit might be. To model organizational units of higher order, we need to introduce labeled hypergraphs.

Definition 5 *Let N be a set of nodes, $\Xi \subseteq \mathcal{P}(N) \times \mathcal{P}(N)$ be a set of hyperedges (whereby $\mathcal{P}(N)$ denotes the power set of N), and $L = L^N \cup L^\Xi$ a set of labels (whereby L^N is a set of node labels and L^Ξ a set of hyperedge labels). Let furthermore $\nu \subseteq N \times L^N$ and $\varepsilon \subseteq \Xi \times L^\Xi$ be two corresponding relations which map nodes and edges to their labels respectively. We denote the tuple $(N, \Xi, L, \nu, \varepsilon)$ as a labeled hypergraph.*

Note that hypergraphs allow us to express higher order relations between sets of organizational units. For example consider a MAS organization that consists of 6 agents $(a_1, ..., a_6)$ as shown on figure 1.

Figure 1. Left: Collaboration Graph; Right: Corresponding Organizational Unit Hypergraph



Agents a_1 and a_2 collaborate together with a common (particular) goal g_1. The same holds for agents a_3 and a_4 with goal g_2, as well as agents a_5 and a_6 with goal g_3. We can model this MAS organization either with the labeled graph on the left if we want to analyze the mutual connections of individual agents, or with the hypergraph on the right to analyze organizational units on a higher level. The organizational unit around objective g_2 can be interpreted in various ways, e.g. as a management team or a virtual overlay unit that integrates the other units.

To model reactive organizational behavior we have to introduce changes in the structure. Changes are modeled with events. From our perspective there are four main types of events that

can occur over time on a graph: (1) update events (due to change of any part of the graph), (2) temporal events (absolute, relative and periodical), (3) implicit events (any event regardless of type that meets a given condition), and (4) complex events (a combination of events constructed using the usual logic operators $\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$).

Definition 6 *With G_i we denote the state of graph G in time i (we assume a discrete time isomorphic to the set of natural numbers $\mathbb{N}$). $\Delta(G_i) = G_{i+1}$ means that G_{i+1} is the result of a changing state in G_i.*

- *An update event $u(G)$ holds in G_i iff $G_i \neq G_{i-1}$. We can classify this type of event further into various changes (insertion, deletion) of various parts of the graph (nodes, edges, labels).*
- *An absolute temporal event $\circledcirc_i(G)$ hold iff G is in state i.*
- *A relative temporal event $\circledcirc_{i+k}(G)$ holds iff G is in state l and there exists a state i such that $i+k=l$.*
- *A periodic temporal event $\circledcirc_{\%j}(G)$ holds iff G is currently in state i and it holds that $i \equiv 0 \% j$*
- *An implicit event $\square_F(G)$ holds iff formula F holds (we read any event that satisfies F).*
- *If e_1 and e_2 are two events then:*
  - *$\neg e_1$ holds if e_1 does not hold.*
  - *$e_1 \wedge e_2$ holds if both e_1 and e_2 hold.*
  - *$e_1 \vee e_2$ holds if either e_1 holds or e_2 holds.*
  - *$e_1 \Rightarrow e_2$ doesn't hold only if e_1 holds but e_2 does not.*
  - *$e_1 \Leftrightarrow e_2$ holds if both $e_1 \Rightarrow e_2$ and $e_2 \Rightarrow e_1$ hold.*

Note that the above definition of events applies both to labeled graphs and subgraphs if edges are replaced with hyperedges. We are now able to introduce AGRR. An AGRR has the ECA (Event-Condition-Action) form borrowed from active database theory, and allows us to state what change has to be done on a graph if a certain event occurs and if a certain condition is met.

Definition 7 *An active graph rewriting rule (AGRR) has the form $E \xrightarrow{C} A$ whereby E is an event, C is a condition formula, and $A = L \rightarrow R$ is a graph transformation. L is called the left-hand-side or pattern graph of the transformation, and R is called the right-hand-side or the replacement graph. An active graph rewriting or active graph grammar (AGG) is a set of AGRRs.*

3. Modeling Example

In order to show an application example of AGG to MAS organization we will create a model of the amoeba organization (Daft, 1992). This organizational structure represents a biomimetic metaphor (Schatten and Žugaj, 2011) in which organizational units are so called amoebas which are autonomous and can split and merge if the number of employees is greater or smaller than given limits respectively.

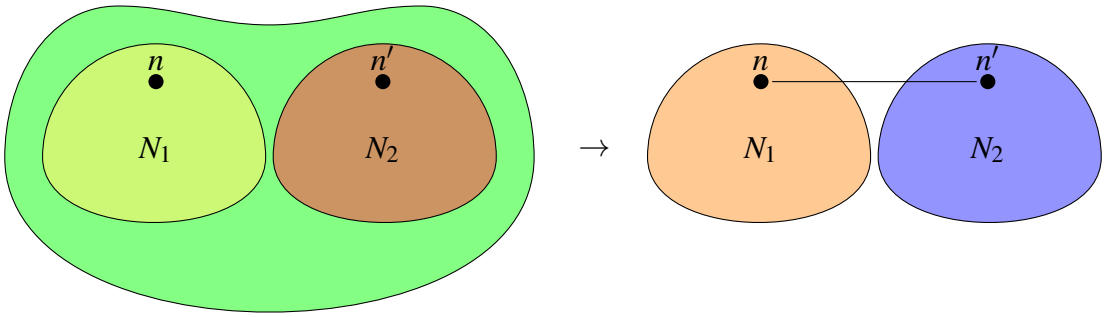
For example, if such an organizational unit acquires more then 100 employees (due to employment of new personnel), the unit will split into two equal amoebas each taking part of the employees. Assume that this organizational unit has been represented by a labeled hypergraph, in which a special node *n* is responsible for tracking the need for a split and is labeled with the label $role : \mathbb{BOSS}$ (the definition of this node is arbitrary, but a token based distributed algorithm might be used).

To model such an organization we can use the following AGGR:

$$u(G) \xrightarrow{(n.count()>100) \wedge (n.role=\mathbb{BOSS})} L_G \rightarrow R_G$$

Whereby $L_G = (N_1 \cup N_2, \{\{n_i | n_i \in N_1 \cup N_2\}\}, L, \nu, \varepsilon)$ and $R_G = (N_1 \cup N_2, \{\{n_i | n_i \in N_1\}, \{n_j | n_j \in N_2\}, \{n, n'\}\}, L, \nu \cup \{(n', role : \mathbb{BOSS})\}, \varepsilon)$. We read on any update event, if a node detects that it has more than 100 collaborators (method $count()$) and if it is the node labeled with $role : \mathbb{BOSS}$, then split the organizational unit connected through the hyperedge $\{n_i | n_i \in N_1 \cup N_2\}$ into two organizational units connected through hyperedges $\{n_i | n_i \in N_1\}$ and $\{n_j | n_j \in N_2\}$. Also connect the responsible $role : \mathbb{BOSS}$ node in one organizational unit with the newly established node n' in the other organizational unit. The split into node sets N_1 and N_2 here is arbitrary, but could be modeled in more detail with additional constraints (to for example take into account that the newly established units have to have all the same roles defined as the original unit etc.). This AGGR can also be represented graphically as shown in figure 2.

Figure 2. AGRR for an Amoeba Organization



On this graphical representation we introduced two arbitrary node sets (N_1 and N_2) on the left-hand-side which are transformed into hyperedges on the right-hand-side.

4. Conclusion & Future Research

In this paper we introduced active graph grammars for modeling organizational structure dynamics in MAS. By borrowing ideas from active database theory we were able to construct a simple graphical formalism that allows us to model MAS organization in a distributed yet (expressively and semantically) powerful way. By introducing a recursive definition of organizational units, we were able to model units and agent roles on any level by using hypergraphs. It has been shown how the usual graph grammars can be extended to account for hypergraphs as well as to work in a dynamic (event-driven) environment.

As opposed to previous studies, the approach outlined herein is holistic since all aspects of change in organizational architecture can be modeled with it: organizational structure, culture, strategy and processes; it naturally corresponds to object-oriented frameworks due to its object-oriented extensions; and allows for specifying change on higher levels of abstraction. Due to its object-oriented extensions it can be easily implemented in object-oriented logic frameworks like frame logic or description logic based systems. Still, since the formalism is highly expressive, its implementation might suffer from combinatoric explosion. Still, since each AGG is local to an agent, there likely won't be complex grammars defined, but this of course has to be tested.

We believe that organizational design of MAS will benefit from a holistic approach to formalizing organizational change. Our future research will target modeling not only changes

in organizational structure, but the other parts of organizational architecture including culture, strategy and processes as well.

References

Barnatt C. (1995) Office space, cyberspace & virtual organization. Journal of General Management, 20(4), 78–91.

Becker B., Beyer D., Giese H., Klein F., and Schilling D. (2006) Symbolic invariant verification for systems with dynamic structural adaptation. In *Proceedings of the 28th international conference on Software engineering*, ICSE '06, pp. 72–81, New York. ACM.

Daft R. L. (1992) Organization Theory and Design. West Publishing Company, Saint Paul etc., (4th ed.).

Fischer T., Niere J., Torunski L., and Zündorf A. (2000) Story Diagrams: A New Graph Rewrite Language Based on the Unified Modeling Language and Java. volume 1764 of Lecture Notes in Computer Science, chapter 21, pages 157–167. Springer.

Giese H., Burmester S., Klein F., Schilling D., and Tichy M. (2003) Multi-agent system design for safety-critical self-optimizing mechatronic systems with uml. In *OOPSLA 2003 - Second International Workshop on Agent-Oriented Methodologies*.

Lesser V., Atighetchi M., Benyo B., Horling B., Raja A., Vincent R., Wagner T., Xuan P., and Zhang S.X.Q. (1999) A Multi-Agent System for Intelligent Environment Control. Computer science technical report, University of Massachusetts.

Nagendra Prasad M. V. and Lesser V. (1999) Learning situation-specific coordination in cooperative multi-agent systems. Autonomous Agents and Multi-Agent Systems, 2(2), 173–207.

Nagendra Prasad M. V., Decker K., Garvey A., and Lesser V. (1996) Exploring Organizational Designs with TAEMS: A case study of distributed data processing. *Proceedings of the Second International Conference on Multi-Agent Systems*, pp. 283–290.

Nagl M. (1979) A tutorial and bibliographical survey on graph grammars. In Volker Claus, Hartmut Ehrig, and Grzegorz Rozenberg, editors, Graph-Grammars and Their Application to Computer Science and Biology, volume 73 of Lecture Notes in Computer Science, pages 70–126. Springer.

Schatten M, and Žugaj, M. (2011) Biomimetics in modern organizations - laws or metaphors? *Interdisciplinary Description of Complex Systems - scientific journal*, 9(1), 39–55.

Smith B., Howard A., Mcnew J.-M., Wang J., and Egerstedt M. (2009) Multi-robot deployment and coordination with embedded graph grammars. Auton. Robots, 26(1), 79–98.

Warnecke H.-J. (1992) Die fraktale fabrik - produzieren im netzwerk (the fractal company - production in the network). In GI Jahrestagung, pages 20–33.

Žugaj M. and Schatten M. (2005) Arhitektura suvremenih organizacija. Tonimir and Faculty of Organization and Informatics, Varaždinske Toplice, Croatia.