Agent-Environment Interaction in a Multi-Agent System: A Formal Model

Gemma Bel-Enguix and M. Dolores Jiménez-López
Research Group on Mathematical Linguistics
Rovira i Virgili University
Pl. Imperial Tarraco, 1
Tarragona, Spain
{gemma.bel,mariadolores.jimenez}@urv.cat

ABSTRACT

In this paper, we introduce a formal-language model for explicitly formalizing agent-environment interaction in a multi-agent systems (MAS) framework: *Conversational Grammar Systems* (CGS). The main goal of the model is to provide a formal framework for defining how agents interact with environments in MAS. CGS offer a model with a high degree of flexibility, what means that they are able to accept new concepts and modify rules, protocols and settings during the computation. Evolution and action are involved in a consistent way in environment, while interaction of agents with the medium is constant. Based on a consolidated and active branch in the field of formal language theory, CGS are a highly formalized framework based in the postulates of artificial life that seems to be quite easy to implement, due to the simplicity of the formalism and the computational background of the theory they use.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*

General Terms

Theory, Languages

Keywords

Eco-Grammar Systems, Artificial Life, Environment, Formal Languages.

1. INTRODUCTION

Many researchers working in the field of multi-agent systems (MASs) agree on the fact that *environments* are essential for multi-agent systems (e.g. [10], [5], [16], [9], [13]). According to [10], *"the environment has been recognized as an explicit and exploitable element to design Multi-Agent Systems (MAS). It can be assigned a number of responsabilities that would be more difficult to design with the sole notion of agents"*. Therefore, we can say that the environment in which a multiagent system is situated is of fundamental importance in the analysis, design and operation of the system. However, according to [16], most researchers neglect to integrate the environment as a primary abstraction in models and tools for MASs, or minimize its responsibilities. Moreover ([5]) while most current multiagent methodologies provide some notion of the environment or the agent's interaction with it, no major methodologies possess a detailed agent-environment interaction model that explicitly defines how the environment is affected by agents or how the agent perceives the environment. As a consequence, a rich potential of applications and techniques that can be developed using MASs is overlooked. In fact, several practical applications have shown how the environment can contribute to manage complex problems. This is why the important role of environment have been emphasized in many models (cf. [13], [16], [9], [1], [5], [14]).

Another important topic in the field of multi-agent systems concerns *formalization*. In [7], it is said that *"while there are many useful models of agents and multi-agent systems, they are typically defined in an informal way and applied in an ad-hoc fashion"*. To develop a formal model that can be used both as the basis of an implementation and also as a precise but general framework is necessary. For some authors ([8]), the use of formalisms is appropriate since they allow unambiguous descriptions of complex systems and also provide proof systems and sets of proof obligations which enable the construction of reliable and robust models. Examples of formal approaches for MAS specifications can be, among others, the models in [6], [8], [7], [3].

The aim of this paper is to *formally* define a multi-agent system that explicitly specifies how an agent interacts with its environment. Therefore, we present an *environment-based* model for formalizing multi-agent systems by means of formal languages. Taking into account the classification of models of environments introduced in [16] in which we can distinguish:

- general models of environments
- inter-agent facilities
- agent-environment interaction
- environments in agent oriented methodologies

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

we can say that the model of environment we introduce in this paper can be classified as an *agent-environment interaction model*, where not only agents but also environment undergo an evolutionary process. Such evolution is brought about mainly by interaction among agents in the environment and by interaction between agents and environment.

The paper is organized as follows: Next section describes in detail our formal-language-theoretic model for multi-agent systems by:

- Identifying the properties of the agents;
- Identifying the properties of the environment;
- Determining the dynamics of the system;
- Identifying the interaction protocol between agents and environment.

First we present the elementary components of the model describing properties of agents and environment. Then, we define elementary configurations of the system. Agent influence on the environment and environmental effects of agents will be also highlighted. Two subsections will be devoted to describe the environmental dynamics and the interaction protocol, respectively. Presentation of the model finishes by defining the output of the system.

Section 3 offers an example that illustrate the functioning of the system. Conclusions and some research lines for future work are provided in Section 4.

Throughout the paper, we assume that the reader is familiar with the basics of formal language theory, for more information see [11].

2. THE FORMAL MODEL

The formal model for multi-agent system we introduce here is based on a field of formal languages called *eco-grammar systems* and introduced in [4]. An eco-grammar system can be defined as an evolutionary multi-agent system where different components, apart from interacting among themselves, interact with a special component called 'environment'. So, within an eco-grammar system we can distinguish two types of components *environment* and *agents*. Both are represented at any moment by a string of symbols that identifies current state of the component. These strings change according to sets of evolution rules (L systems). Interaction among agents and environment is carried out through agents' actions performed on the environmental state by the application of some productions (rewriting rules) from the set of action rules of agents. The concept of eco-grammar system is based on six postulates formulated according to properties of artificial life:

1. An ecosystem consists of an *environment* and a *set of agents*. Both state of the environment and states of agents are described by strings of symbols of given alphabets.
2. In an ecosystem there is a *universal clock* which marks time units, the same for all the agents and for the environment, according to which agents and environment evolution is considered.
3. Both environment and agents have characteristic *evolution rules* which are in fact L systems, hence are applied in a parallel manner to all the symbols describing agents and environment; such a (rewriting) step is done in each time unit.
4. Evolution rules of environment are *independent* on agents and on the state of the environment itself. Evolution rules of agents *depend on* the state of the environment (at a given moment, a subset of applicable rules is chosen from a general set associated to each agent).
5. Agents act on the environment according to *action rules*, which are pure rewriting rules used sequentially. In each time unit, each agent uses one action rule which is chosen from a set depending on current state of the agent.
6. *Action has priority over evolution* of the environment. At a given time unit exactly the symbols which are not affected by action (in the environment) are rewritten (in a parallel manner) by evolution rules.

We have modified basic definitions of eco-grammar systems by taking closely into account one of the most common forms of interaction: *dialogue*. The result is a new formal model we have called *Conversational Grammar Systems* (CGS). CGS offer a framework with a high degree of flexibility, what means that they are able to accept new concepts and modify rules, protocols and settings during the computation. Evolution and action are involved in a consistent way in environment/contexts, while interaction of agents with the medium is constant.

2.1 A Multi-Agent System: CGS

DEFINITION 1. *A Conversational Grammar System (CGS) of degree n, $n \geq 2$, is an $(n+1)$-tuple:*

$$\Sigma = (E, A_1, ..., A_n),$$

where:

- $E = (V_E, P_E)$,
  - V_E *is an alphabet;*
  - P_E *is a finite set of rewriting rules over* V_E.
- $A_i = (V_i, P_i, R_i, \varphi_i, \psi_i, \pi_i, \rho_i)$, $1 \leq i \leq n$,
  - V_i *is an alphabet;*
  - P_i *is a finite set of rewriting rules over* V_i;
  - R_i *is a finite set of rewriting rules over* V_E;
  - φ_i: $V_E^* \to 2^{P_i}$;
  - ψ_i: $V_E^* \times V_i^+ \to 2^{R_i}$;
  - π_i *is the start condition;*
  - ρ_i *is the stop condition;*
  - π_i *and* ρ_i *are predicates on* V_E^*. *We can define the following special types of predicates. We say that predicate* σ *on* V_E^* *is of:*
    - *Type (a) iff* $\sigma(w) =$ *true for all* $w \in V_E^*$;

* *Type (rc) iff there are two subsets R and Q of V_E and $\sigma(w) = true$ iff w contains all letters of R and w contains no letter of Q;*
* *Type (K) iff there are two words x and x' over V_E and $\sigma(w) = true$ iff x is a subword of w and x' is not a subword of w;*
* *Type (K') iff there are two finite subsets R and Q of V_E^* and $\sigma(w) = true$ iff all words of R are subwords of w and no word of Q is a subword of w;*
* *Type (C) iff there is a regular set R over V_E and $\sigma(w) = true$ iff $w \in R$.*

2.2 Properties of Agents

According to [5], an agent is anything that can sense and perform actions upon its environment.

In our formal model, an agent (A_i, $1 \leq i \leq n$) is identified at any moment by a string of symbols w_i, over alphabet V_i, which represents its current state. This state can be changed by applying evolution rules from P_i, which are selected according to mapping φ_i and depend on the state of the environment.

A_i can modify the state of the environment by applying some of its action rules from R_i, which are selected by mapping ψ_i and depend both on the state of the environment and on the state of the agent itself.

Start/Stop conditions of A_i are determined by π_i and ρ_i, respectively. A_i starts/stops its actions if context matches π_i and ρ_i. Start/stop conditions of A_i can be of different types: (a) states that an agent can start/stop at any moment. (rc) means that it can start/stop only if some letters are present/absent in the current sentential form. And (K), (K') and (C) denote such cases where global context conditions have to be satisfied by the current sentential form.

2.3 Properties of the Environment

According to [16], in situated MASs, the environment is an active entity with its own processes that can change its own state, independent of the activity of the embedded agents. In our model, E represents the environment described at any moment of time by a string w_E, over alphabet V_E, called the *state of the environment*. The state of the environment is changed both by its own evolution rules P_E and by the actions of the agents of the system, A_i, $1 \leq i \leq n$.

Russell and Norvig [12] discuss a number of key properties of environments that are now adopted by most researchers in the domain:

* Accessible versus inaccessible: indicates whether the agents have access to the complete state the environment or not.
* Deterministic versus nondeterministic: indicates whether a state change of the environment is uniquely determined by its current state and the actions selected by the agents or not.
* Static versus dynamic: indicates whether the environment can change while an agent deliberates or not.
* Discrete versus continuous: indicates whether the number of percepts and actions are limited or not.

According to this properties, we could classified environment of CGS as accessible, deterministic, dynamic and discrete.

2.4 Elementary Configurations

In CGS, we define an elementary configuration as a *state* in which the system can be at a given time.

Definition 2. *A state of a CGS $\Sigma = (E, A_1, \ldots, A_n)$, $n \geq 2$, is an $n+1$-tuple:*

$$\sigma = (w_E; w_1, \ldots, w_n),$$

where $w_E \in V_E^*$ is the state of the environment, and $w_i \in V_i^*$, $1 \leq i \leq n$, is the state of agent A_i.

2.5 Agent Influence on the Environment

Agent influence on the environment in CGS is carried out through *actions* that agents perform on the environmental string. The behaviour of the system in CGS is described as a sequence of *context-change-actions* allowed by the current environment and performed by two or more *agents.*

Different definitions of *action* can be found in the literature of MASs:

* According to [16], the classical approach to deal with actions is based on the (environmental) transformation of states, i.e. an action is defined as a transition state, that is, as an operator whose execution produces a new state. From an observational point of view, the result of the behavior or an agent -its action- is directly modelled by modifying the environmental state variables.
* For [5], an action is defined as an entity that represents the agent's actual sensor or effector. Specification of the execution of an action is defined via a single accessible operation. Each action's operation has a set of preconditions that determine whether or not the operation can be executed. If the preconditions hold the operation may be executed.
* In [8], an action is a discrete event which change the state of the environment.

In CGS, an *action* is defined as the application of a rule *on the environmental string*. This rule is applied to the state of the environment *by an active agent*, and it is not any rule, but a rule selected by $\psi_i(w_E, w_i)$, that is, a rule (an action) allowed by the current context and by the state of the agent itself. We define an *active agent* in relation to the allowable actions it has at a given moment. Formally:

Definition 3. *By an action of an active agent A_i in state $\sigma = (w_E; w_1, w_2, \ldots, w_n)$ we mean a direct derivation step performed on the environmental state w_E by the current action rule set $\psi_i(w_E, w_i)$ of A_i.*

Definition 4. *An agent A_i is said to be active in state $\sigma = (w_E; w_1, w_2, \ldots, w_n)$ if the set of its current action rules, that is, $\psi_i(w_E, w_i)$, is a nonempty set.*

2.6 Environmental Effects on Agents

Conversational grammar systems can be defined as an environment-based model for multi-agent systems. Up to now we have seen that everything in our model depend on the environment: the set of agents that form the system share a common environment and performs actions on it;

these actions are selected by function that takes into account the state of the environment. However, environment in CGS not only constraint agents' environment, but it also has a direct effect on agents' strings. In fact, during the course of the computation, agents' states are modified through the application of evolution rules that are selected by a mapping that takes into account only the state of the environment. Formally,

DEFINITION 5. *Let* $\sigma = (w_E; w_1, \ldots, w_n)$ *and* $\sigma' = (w'_E; w'_1, \ldots, w'_n)$ *be two states of a CGS. We say that* σ' *arises from* σ *by an evolution step, denoted by* $\sigma \overset{e}{\Longrightarrow}_\Sigma \sigma'$, *iff the following conditions hold:*

- w'_E *can be directly derived from* w_E *by applying rewriting rule set* P_E;
- w'_i *can be directly derived from* w_i *by applying rewriting rule set* $\varphi_i(w_E)$, $1 \leq i \leq n$.

2.7 Environmental Dynamics

Environmental dynamics in CGS is understood in terms of *context changes*. In order to formalize how the environment passes from one state to another as a result of agents' actions we introduce the following definition:

DEFINITION 6. *Let* $\sigma = (w_E; w_1, \ldots, w_n)$ *and* $\sigma' = (w'_E; w'_1, \ldots, w'_n)$ *be two states of a CGS. We say that* σ' *arises from* σ *by a simultaneous action of active agents* $A_{i_1}, \ldots, A_{i_r}$, *where* $\{i_1, \ldots, i_r\} \subseteq \{1, \ldots, n\}$, $i_j \neq i_k$, *for* $j \neq k$, $1 \leq j, k \leq r$, *onto the state of the environment* w_E, *denoted by* $\sigma \overset{a}{\Longrightarrow}_\Sigma \sigma'$, *iff:*

- $w_E = x_1 x_2 \ldots x_r$ *and* $w'_E = y_1 y_2 \ldots y_r$, *where* x_j *directly derives* y_j *by using current rule set* $\psi_i(w_E, w_{i_j})$ *of agent* A_{i_j}, $1 \leq j \leq r$;
- *there is a derivation:*

$$w_E = w_0 \overset{a}{\Longrightarrow}^*_{A_{i_1}} w_1 \overset{a}{\Longrightarrow}^*_{A_{i_2}} w_2 \overset{a}{\Longrightarrow}^*_{A_{i_3}} \ldots \overset{a}{\Longrightarrow}^*_{A_{i_r}} w_r = w'_E$$

  such that, for $1 \leq j \leq r$, $\pi_{i_j}(w_{j-1}) = true$ *and* $\rho_{i_j}(w_j) = true$. *And for* $f \in \{t, \leq k, \geq k\}$ *the derivation is:*

$$w_E = w_0 \overset{a}{\Longrightarrow}^f_{A_{i_1}} w_1 \overset{a}{\Longrightarrow}^f_{A_{i_2}} w_2 \overset{a}{\Longrightarrow}^f_{A_{i_3}} \ldots \overset{a}{\Longrightarrow}^f_{A_{i_r}} w_r = w'_E$$

  such that, for $1 \leq j \leq r$, $\pi_{i_j}(w_{j-1}) = true$[1], *and*
- $w'_i = w_i$, $1 \leq i \leq n$.

However, as we have already said, the environment in our model is an active entity with its own processes that can change its own state, independent of the activity of the agents. In order to formalize this we recall again the following definition that make explicit how evolution rules of the environment are applied:

DEFINITION 7. *Let* $\sigma = (w_E; w_1, \ldots, w_n)$ *and* $\sigma' = (w'_E; w'_1, \ldots, w'_n)$ *be two states of a CGS. We say that* σ' *arises from* σ *by an evolution step, denoted by* $\sigma \overset{e}{\Longrightarrow}_\Sigma \sigma'$, *iff the following conditions hold:*

- w'_E *can be directly derived from* w_E *by applying rewriting rule set* P_E;
- w'_i *can be directly derived from* w_i *by applying rewriting rule set* $\varphi_i(w_E)$, $1 \leq i \leq n$.

2.8 Interaction Protocol

According to [5], by executing an operation defined in an action, an agent can sense or manipulate its environment. In other words, by performing an action, agents interact with the environment. Coordination is defined in many ways but in its simplest form it refers to ensuring that the actions of independent actors (agents) in an environment are coherent in some way. The challenge, therefore, is to identify mechanisms that allow agents to coordinate their actions. Research to date has identified a huge range of different types of coordination and cooperation mechanisms, raging from emergent cooperation, coordination protocols to distributed planning. In CGS, we define different *modes of derivation* that can be seen as the interaction protocol of our multi-agent system:

DEFINITION 8. *Let* $\Sigma = (E, A_1, ..., A_n)$ *be a CGS. And let* $w_E = x_1 x_2 ... x_r$ *and* $w'_E = y_1 y_2 ... y_r$ *be two states of the environment. Let us consider that* w'_E *directly derives from* w_E *by action of active agent* A_i, $1 \leq i \leq n$, *as shown in Definition 6. We write that:*

$$w_E \overset{a}{\Longrightarrow}^{\leq k}_{A_i} w'_E \text{ iff } w_E \overset{a}{\Longrightarrow}^{\leq k'}_{A_i} w'_E, \text{ for some } k' \leq k;$$

$$w_E \overset{a}{\Longrightarrow}^{\geq k}_{A_i} w'_E \text{ iff } w_E \overset{a}{\Longrightarrow}^{\leq k'}_{A_i} w'_E, \text{ for some } k' \geq k;$$

$$w_E \overset{a}{\Longrightarrow}^{*}_{A_i} w'_E \text{ iff } w_E \overset{a}{\Longrightarrow}^{k}_{A_i} w'_E, \text{ for some } k;$$

$$w_E \overset{a}{\Longrightarrow}^{t}_{A_i} w'_E \text{ iff } w_E \overset{a}{\Longrightarrow}^{*}_{A_i} w'_E \text{ and there is no } z \neq y \text{ with } y \overset{a}{\Longrightarrow}^{*}_{A_i} z.$$

In words, $\leq k$-derivation mode represents a time limitation where A_i can perform at most k successive actions on the environmental string. $\geq k$-derivation mode refers to the situation in which A_i has to perform at least k actions whenever it participates in the derivation process. With $*$-mode, we refer to such situations in which agent A_i performs as many actions as it wants to. And finally, t-derivation mode represents such cases in which A_i has to act on the environmental string as long as it can.

One way of getting transitions with no gap and no overlap in CGS is to endow agents with an *internal control* that contains start/stop conditions that allow agents to recognize places where they can start their activity, as well as places where they should stop their actions and give others the chance to act. This is, start/stop conditions help agents to recognize *transition relevance places*, i.e. places where speaker change occurs. Start/stop conditions have been formally defined in Definition 1.

2.9 Output of the System

In CGS, computation implies that both the *state of the environment* and *state of agents* change. Such changes take place thanks to two different types of processes: *action steps* and *evolution steps*. By means of the former, active agents perform actions on the environmental string modifying its state; the latter imply the reaction of context and agents

[1]In this latter case the stop condition $\rho_i(w_j)$ = true is replaced by the stop condition given the f-mode.

which, according to the changes produced by agents' actions, modify their states. So, action steps and evolution steps alternate in the course of the computation. At the end, what we have is a *sequence of states* reachable from the initial state by performing, alternatively, action and evolution derivation steps:

Definition 9. *Let $\Sigma = (E, A_1, \ldots, A_n)$ be a CGS and let σ_0 be a state of Σ. By a state sequence (a derivation) starting from an initial state σ_0 of Σ we mean a sequence of states $\{\sigma_i\}_{i=0}^{\infty}$, where:*

- $\sigma_i \overset{a}{\Longrightarrow}_{\Sigma} \sigma_{i+1}$, *for $i = 2j$, $j \geq 0$; and*
- $\sigma_i \overset{e}{\Longrightarrow}_{\Sigma} \sigma_{i+1}$, *for $i = 2j+1$, $j \geq 0$.*

Definition 10. *For a given CGS Σ and an initial state σ_0 of Σ, we denote the set of state sequences of Σ starting from σ_0 by $Seq(\Sigma, \sigma_0)$.*

The set of environmental state sequences is:

$Seq_E(\Sigma, \sigma_0) = \{\{w_{Ei}\}_{i=1}^{\infty} \mid \{\sigma_i\}_{i=0}^{\infty} \in Seq(\Sigma, \sigma_0),$
$\sigma_i = (w_{Ei}; w_{1i}, \ldots, w_{ni})\}.$

The set of state sequences of the j-th agent is defined by:

$Seq_j(\Sigma, \sigma_0) = \{\{w_{ji}\}_{i=1}^{\infty} \mid \{\sigma_i\}_{i=0}^{\infty} \in Seq(\Sigma, \sigma_0),$
$\sigma_i = (w_{Ei}; w_{1i}, \ldots, w_{ji}, \ldots, w_{ni})\}.$

$Seq(\Sigma, \sigma_0)$ describes the behavior of the system, this is, the possible state sequences, directly following each other, starting from the initial state. $Seq_E(\Sigma, \sigma_0)$ and $Seq_j(\Sigma, \sigma_0)$ are the corresponding sets of sequences of the states of the environment and of the states of j-th agent, respectively.

Now, we associate certain languages with an initial configuration:

Definition 11. *For a given CGS Σ and an initial state σ_0 of Σ, the language of the environment is:*

$L_E(\Sigma, \sigma_0) = \{w_E \in V_E^* \mid \{\sigma_i\}_{i=0}^{\infty} \in Seq(\Sigma, \sigma_0),$
$\sigma_i = (w_E; w_1, \ldots, w_n)\}.$

and the language of j-th agent is:

$L_j(\Sigma, \sigma_0) = \{w_j \in V_A^* \mid \{\sigma_i\}_{i=0}^{\infty} \in Seq(\Sigma, \sigma_0),$
$\sigma_i = (w_E; w_1, \ldots, w_j, \ldots, w_n)\}.$

for $j = 1, 2, \ldots, n$.

$L_E(\Sigma, \sigma_0)$ and $L_j(\Sigma, \sigma_0)$ correspond to those states of the environment and to those states of the j-th agent, respectively, that are reachable from the initial configuration of the system.

Figure 1 gives a graphic idea of CGS.

3. EXAMPLE

Example 1. *Consider the following CGS:*
$\Sigma = (E, A_1, A_2)$, where:

- $E = (V_E, P_E)$,
  - $V_E = \{a, x, y\}$;
  - $P_E = \{a \to b^2, b \to a^2, x \to x, y \to y\}$.

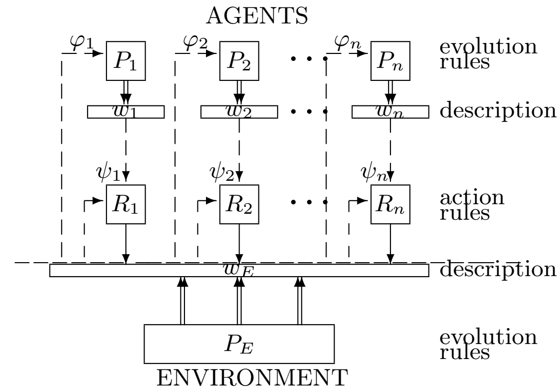

Figure 1: Conversational Grammar Systems.

- $A_1 = (V_1, P_1, R_1, \varphi_1, \psi_1, \pi_1, \rho_1)$ *with:*
  - $V_1 = \{c\}$;
  - $P_1 = \{c \to c\}$; $R_1 = \{a \to x\}$;
  - $\varphi_1(w) = P_1$ *for every* $w \in V_E^*$;
  - $\psi_1(w; u) = R_1$ *for* $w \in \{a, x, y\}^*$ *and* $u = c$, *otherwise* $\psi_1(w; u) = \emptyset$;
  - $\pi_1 =$ *true for all* $w \in V_E^*$; $\rho_1 =$ *true for all* $w \in V_E^*$.
- $A_2 = (V_2, P_2, R_2, \varphi_2, \psi_2, \pi_2, \rho_2)$ *with:*
  - $V_2 = \{d\}$;
  - $P_2 = \{d \to d\}$; $R_2 = \{b \to y\}$;
  - $\varphi_2(w) = P_2$ *for every* $w \in V_E^*$;
  - $\psi_2(w; v) = R_2$ *for* $w \in \{b, x, y\}^*$ *and* $v = d$, *otherwise* $\psi_2(w; v) = \emptyset$;
  - $\pi_2 =$ *true for all* $w \in V_E^*$; $\rho_2 =$ *true for all* $w \in V_E^*$.

P_E, P_1 and P_2 contain rules of an 0L system applied in a parallel way. Rules in R_1 and R_2 are pure context-free productions applied sequentially. Let us suppose that the system is working in the arbitrary mode $*$. And let us take $\sigma_0 = (a^3; c, d)$ as the initial state of Σ. Then, a possible derivation in Σ is the following one:

$$(a^3; c, d) \overset{a}{\Longrightarrow}{}_{\Sigma}^{*} (a^2x; c, d) \overset{e}{\Longrightarrow}{}_{\Sigma}^{*} (b^4x; c, d) \overset{a}{\Longrightarrow}{}_{\Sigma}^{*}$$
$$(yb^3x; c, d) \overset{e}{\Longrightarrow}{}_{\Sigma}^{*} (ya^6x; c, d) \overset{a}{\Longrightarrow}{}_{\Sigma}^{*} (ya^2xa^3x; c, d) \overset{e}{\Longrightarrow}{}_{\Sigma}^{*} \ldots$$

Notice, that we alternate action and evolution steps. At every action step one of the agents rewrites one symbol of the environmental state, while in evolution steps both environmental and agents' states are rewritten according to 0L rules.

In this section, we have just offered a formal example of how CGS works. Space limitations prevent us from offering a more illustrative example of CGS possible applications. The reader can see [2] for a more detailed application of the model. In [2], by combining CGS with the Multi-Agent Protocol (MAP) introduced in [15] language, we present a simple formal device that could be used for the design of dialogue systems with limited human-like behaviour.

CGS offers a model with a high degree of flexibility where strings can be modified during running time, allowing to dynamically alter agent behaviour according to the context changes. Taking into account that human-computer interfaces require models of dialogue structure that capture the variability and unpredictability within dialogue, we claim that CGS may offer a useful formalism for the field of computer dialogue systems. CGS may be used as a formal-language interaction protocol for agent communication and may contribute to the building of better human-computer dialogues through a simulation of human language use.

4. FINAL REMARKS

Many authors have emphasized the importance of environment in multi-agent systems. In [5], it is said that all multi-agent systems methodologies should provide a robust way to define the interaction of agents with their environments. According to [14], the concept of an environment for multi-agent systems is being recognized as a promising research area. The research on MAS environments originates from realizing the benefits and potential of making the environment explicit. Most agent-oriented design methodologies focus on goals and decision processes and neglect the environment explicitly. In this paper, we have introduced a formal framework for modelling multi-agent system's interactions with its environment. CGS can be defined as an environment-based model due to the fact that everything in the system depends on the environment which controls actions agents can perform, changes the state of the agents and contains the output of the system.

One of the main advantages of our model is that it is presented in formal and non ambiguous terms. According to [8], formalization provides clarity in characterizing the nature of concepts. There is a demand of formal modelling with the need for implementation by providing clear and unambiguous definitions of state and operations on state which provide the basis for program development. We need to be formal to be precise about the concepts we discuss, yet we want to remain directly connected to issues of implementation. Conversational Grammar Systems are based on a consolidated and active branch in the field of formal language theory, so they offer a highly formalized framework that seems to be quite easy to implement, due to the simplicity of the formalism and the computational background of the theory we use. Achieving a valid and simple computational implementation of this model is the major research line for the future.

5. REFERENCES

[1] M. Bartlett and D. Kazakov. The role of environment structure in multi-agent simulations of language evolution. In *Proceedings of the Fourth Symposium on Adaptive Agents and Multi-Agent Systems*, 2004.

[2] G. Bel-Enguix, A. Grando, and M.D. Jiménez-López. A grammatical framework for modlling multi-agent dialogues. In *Shi, Z. & Sadananda, R. (eds.), Agent Computing and Multi-Agent Systems: 9th Pacific Rim International Workshop on Multi-Agents, PRIMA 2006*, page 10Ũ21. Lecture Notes in Computer Science (LNCS), 4088, Springer, Berlin, 2006.

[3] D. Capera, J. Fanchon, J.-P. George, and V. Camps. A generic model based on automata for multi-agent systems. In *EUMAS 2005*, pages 79–89. KVAB, Brussels, 2005.

[4] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, and G. Păun. Eco-grammar systems: A grammatical framework for life-like interactions. *Artificial Life*, 3(1):1–28, 1996.

[5] S. A. DeLoach and J. L. Valenzuela. An agent-environment interaction model. In *L. Padgham and F. Zambonelli (eds.), AOSE 2006*, pages 1–18. LNCS 4405, Springer, Berlin, 2007.

[6] V. Hilaire, P. Gruer, A. Koukam, and O. Simonin. Formal specification approach of role dynamics in agent organizations: Applications to the satisfaction-altruism model. *International Journal of Software Engineering and Knowledge Engineering*, to appear.

[7] V. Hilaire, O. Simonin, A. Koukam, and J. Ferber. A formal approach to design and reuse agent and multiagent models. In *Agent Oriented Software Engineering*, pages 142–157. LNCS 3382, Springer, Berlin, 2004.

[8] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 254–260. AAAI Press/MIT Press, 1995.

[9] A. Ominici, A. Ricci, M. Viroli, C. Castelfranci, and L. Tummolini. Coordination artifacts: Environment-based coordination for intelligen agents. In *Proceedings of 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 286–293. ACM, New York, 2004.

[10] E. Platon, M. Mamei, N. Sabouret, S. Honiden, and H. V. D. Parunak. Mechanisms for environments in multi-agent systems. survey and opportunities. *Autonomous Agents & Multi-Agent Systems*, 14(1), 2007.

[11] G. Rozenberg and A. Salomaa. *Handbook of Formal Languages*. Springer, Berlin, 1997.

[12] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[13] O. Simonin and F. Gechter. An environment-based methodology to design reactive multi-agent systems for problem solving. In *Environments for Multi-Agent Systems II*, pages 32–49. LNAI 3830, Springer, Berlin, 2006.

[14] P. Valckenaers, J. Sauter, C. Sierra, and J. A. Rodríguez-Aguilar. Applications and environments for multi-agent systems. *Autonomous Agents and Multi-Agents Systems*, 14:61–85, 2007.

[15] C. Walton. Multi-agent dialogue protocols. In *Proceedings of the Eighth International Symposium in Artificial Intelligence and Mathematics*. Fort Lauderdale, Florida, 2004.

[16] D. Weyns, H. V. D. Parunak, F. Michel, T. Holvoet, and J. Ferber. Environments for multiagent systems. state-of-the-art and research challenges. In *E4MAS 2004*, pages 1–47. LNAI 3374, Springer, Berlin, 2005.