Dialogue as Inter-action

Gemma Bel-Enguix and M. Dolores Jiménez-López

GRLMC-Research Group on Mathematical Linguistics
Rovira i Virgili University
Pl. Imperial Tarraco, 1
43005 Tarragona, Spain

Abstract. In this paper we introduce a *formal model* of dialogue based on grammar systems theory: *Conversational Grammar Systems* (CGS). The model takes into account ideas from the study of human-human dialogue in order to define a flexible mechanism for coherent dialogues that may help in the design of effective and user-friendly computer dialogue systems. The main feature of the model is to present an *action view* of dialogue. CGS model dialogue as an inter-action, this is a sequence of *acts* performed by two or more agents in a common environment. We claim that CGS are able to model dialogue 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.

1 Introduction

Human-computer interaction (HCI) did not exist as a field of scientific inquiry in the earliest days of computers because very few people interacted with computers, and those who did generally were technical specialists. Papers on the topic began to appear only in the 1960s. As more and more people found themselves using computers for a broadening variety of tasks, the topic became an important focus of research. HCI has now been a major area of research in computer science, human factors, engineering psychology and closely related disciplines.

According to [12], a goal of human factors research with computer systems is to develop human-computer communication modes that are both error tolerant and easily learned. Since people already have extensive communication skills through their own native or natural language, many believe that natural language interfaces can provide the most useful and efficient way for people to interact with computers. Taking into account this idea, what we propose in this paper is to start by the study and analysis of spoken human-human dialogues in order to abstract their main principles and mechanisms and to apply them to the definition of a formal model of dialogue that may be used for designing effective, efficient and user-friendly computer dialogue systems.

Research on dialogue has been largely absent from academic disciplines till the second half of this 20th century. Importance of dialogue was discovered by an empirical discipline, known as Conversation Analysis, that emerged in the early 1960s within the field of ethnomethodology. The main purpose of that research stream –always related to the names of Sacks, Schegloff and Jefferson– can be stated quite simple: to describe

Bel-Enguix G. and Dolores Jiménez-López M. (2007).
Dialogue as Inter-action.
In *Proceedings of the 4th International Workshop on Natural Language Processing and Cognitive Science*, pages 181-190
DOI: 10.5220/0002418001810190

'technology of dialogue.' The most important objective of conversation analysis is to explain procedures used by participants in a dialogue to produce utterances and to make sense of other people's talk. Being concerned with talk as a collaborative matter and with how parties can jointly produce an organised sequence of talk, conversation analysis tries to specify how the consecutive actions that dialogue consists of are related one to another and how they build up a conversational sequence. Methodology and results obtained by researchers in the field of conversation analysis have revealed as quite useful in the area of human-computer interaction. Computer scientists such as Norman and Thomas or Robinson have pointed out that utility:

> '*Conversation Analysis seems to us to offer the possibility of the provision of comprehensive and secure design information based on a coherent view of interaction, although representing an investigative paradigm quite different to those currently employed in Human-Computer Interaction research.*' [11].

> '*The findings of one particular form of ethnomethodological work, that of Conversational Analysis, seem prima facie to be directly relevant to human-computer interaction.*' [14].

The model we introduce in this paper is based on the theory of grammar systems and takes into account ideas from the study of human-human dialogue in order to define a flexible mechanism for coherent dialogues. The aim of the model is to see how productive can be to reproduce in human-computer dialogues details of natural conversations between people. In order to fulfil that goal, we use in our formal model ideas, techniques and procedures that have been proposed to account for human dialogue. The main feature of this model is to present an *action view* of dialogue. Therefore, next section will be devoted to overview some action-based approaches to language. Section 3 will introduce Conversational Grammar Systems as a formal-language-model that defines dialogue as Inter-Action. Last section present some final remarks and directions for future work.

Throughout the paper, we assume that the reader is familiar with the basics of formal language theory, for more information see [15].

2 Dialogue as Inter-action

Within a philosophical tradition begun by Austin [3], dialogue is viewed as a sequence of *speech acts*, uttered by each party to achieve certain goals. He observes that there exists a type of utterances that do not describe or report anything at all, but that their uttering is the *doing of an action*. He calls this special type of utterances *performative sentences*, in order to stress the idea that the issuing of the utterance is the performing of an action and not just the saying of something (as is the case of constative sentences). After having postulated the existence of performative sentences that cannot be said to be 'true' or 'false,' but that can be qualified at most as 'happy' or 'unhappy,' Austin observes that in any utterance we can individuate three different types of acts, being one of them the act of *doing* something while uttering the sentence.

By introducing the idea of illocutionary act, Austin opens what has been a very influential theory, namely theory of *speech acts*. However, that theory would not have had the repercussion it has actually had without the figure of John Searle. Work done by this author in the field of speech acts is considered as the systematic development and continuation of Austin's

Speech act theory considers the interactive use of language to be of primary importance. According to [6], speech act theory has been a major source of inspiration for all action-based approaches to language, and has been fruitful both in the development of pragmatics and as a conceptual framework for thinking about human computer dialogue. *Action view* of dialogue is perfectly resumed in Searle's sentence: '*Talking is performing acts according to rules.*' [16].

An important action theory of language is *Dynamic Interpretation Theory* (DIT) introduced by Bunt [5]. In DIT, dialogues are viewed in an *action perspective*. Language is considered a tool to perform context-changing actions. According to DIT, a dialogue can be analysed in terms of combinations of actions called *dialogue acts* defined as: '*Functional units used by the speaker to change the context.*' [5].

Many authors have defended the idea that to use language is to perform *acts* according to rules. Next to speech act theory or Bunt's DIT, we can find thesis as the one presented by Clark [7] who views language use –and, therefore, dialogue– as a *joint action*, defining joint action as '*One that is carried out by an ensemble of people acting in coordination with each other.*' For this researcher, '*What people do in arenas of language use is to take actions.*'.

In a similar fashion, in [9] it is claimed that analysis of dialogue ought to be based on a theory of collective action. These authors take language as action and study those aspects of language use which can be explained following general principles of cooperative interaction. Also for Sharrock & Anderson, the primary characteristic of the utterances conversation analysis deals with is '*often less that they are verbal actions, but that they are actions.*' [17].

Another general action-based approach to language has been developed by Allwood and co-workers and has been called *Communicative Activity Analysis* [1] [2]. Like speech act theory, Allwood's approach takes the view that communication is action and provides a conceptual analysis of action, social activity and ethics in communication with considerable depth and generality.

The above are just few examples of the *action view* of language use. All of them share the idea of defining a dialogue as to *perform actions in a specific context.* In order to apply these ideas to the design of dialogue systems, we need a formalized theory that takes into account both general principles of natural language dialogue and, of course, this generalized view of dialogue as action. In the next section, we provide a formal model that tries to capture these ideas by using a formal-language-theoretical framework.

3 Conversational Grammar Systems: An Inter-action Model

The model we introduce here is based on Grammar Systems Theory. Grammar systems can be characterized as a device where agents *perform actions according to rules.*

Grammar systems theory is a consolidated and active branch in the field of formal languages [8] that provides syntactic models for describing multi-agent systems at the symbolic level, using tools from formal grammars and languages. Grammar systems theory has been widely investigated and nowadays constitutes a well-developed formal theory that presents several advantages with respect to classical models. However, being a branch of formal languages, researchers in the field of grammar systems have concentrated mainly on theoretical aspects. Roughly speaking, a grammar system is a *set* of grammars working together, according to a specified protocol, to generate a language. Notice that while in classical formal language theory *one* grammar (or automata) works individually to generate (or recognize) *one* language; here, instead, we have *several* grammars working together in order to produce *one* language.

While grammar systems are related to Artificial Intelligence, a subfield of the theory, –the so-called eco-grammar systems– is closely related to Artificial Life. Eco-grammar systems provide a syntactical framework for eco-systems, this is, for communities of evolving agents and their interrelated environment. Briefly, an eco-grammar system is defined as a multi-agent system where different components, apart from interacting among themselves, interact with a special component called 'environment' [13].

Here we introduce a new model: *Conversational Grammar Systems* (CGS). CGS are multi-agent systems based on grammar systems, specifically in the so-called eco-grammar systems. Conversational grammar system 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 inter-action of agents with the medium is constant.

According to the idea that dialogue '*can be understood as the sustained production of chains of mutually-dependent acts, constructed by two or more agents each monitoring and building on the actions of the other.*' [10], conversational grammar systems intend to describe dialogue as a sequence of *context-change-actions* allowed by current environment and performed by two or more *agents*. Therefore, in conversational grammar systems we understand dialogue as inter-action, this is a sequence of *acts* performed by two or more agents in a common environment.

In what follows we introduce the formal definition of our model.

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*: an alphabet;*
  - P_E*: a finite set of rewriting rules on* 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*: an alphabet;*
  - P_i*: a finite set of rewriting rules on* V_i;

- R_i*: a finite set of rewriting rules on* V_E;
- φ_i: $V_E^* \to 2^{P_i}$;
- ψ_i: $V_E^* \times V_i^+ \to 2^{R_i}$;
- π_i*: the start condition;*
- ρ_i*: the stop condition;*
- π_i *and* ρ_i*: 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* $w \in R$.

The items of the above definition have been interpreted as follows: a) 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, $1 \leq i \leq n$. b) A_i, $1 \leq i \leq n$, represents an agent. It 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.

CGSs intend to describe dialogue as a sequence of *context-change-actions* allowed by the current environment and performed by two or more *agents*. In this view, an *action* is defined as the application of a rule *on the environmental string*:

Definition 2 *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 3 *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.

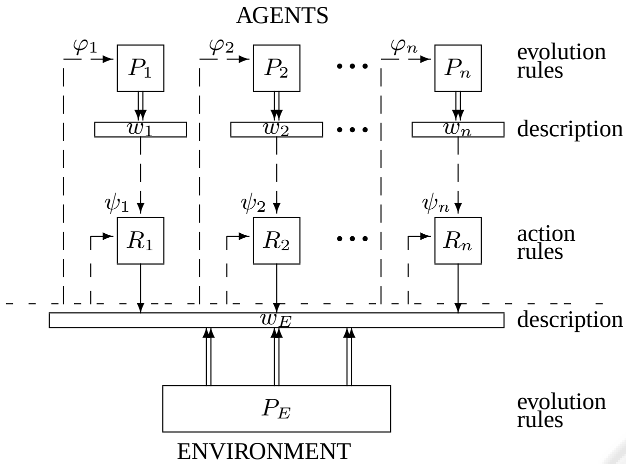

Fig. 1. Conversational Grammar Systems.

This rule is applied *by an active agent* and it is a rule selected by $\psi_i(w_E, w_i)$.

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.*

Since dialogue in CGS is understood in terms of *context changes* , we have to define how the environment passes from one state to another as a result of agents' actions:

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 $\Sigma = (E, A_1, \ldots, A_n)$. 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
- *$w'_i = w_i, 1 \leq i \leq n$.*

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

However, in the course of a dialogue, agents' states are also modified and the environmental string is subject to changes due to reasons different from agents' actions. So, in order to complete our formalization of dialogue development, we add 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 $\Sigma = (E, A_1, \ldots, A_n)$. We say that σ' arises from σ by an evolution step, denoted by $\sigma \stackrel{e}{\Longrightarrow}_\Sigma \sigma'$, 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 $\varphi_i(w_E)$, $1 \leq i \leq n$.*

In CGS, the development of dialogue 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 which, according to the changes produced by agents' actions, modify their states. So, action steps and evolution steps alternate in the course of dialogue. 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 7 *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:*

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

Definition 8 *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:
$Seq_j(\Sigma, \sigma_0) = \{\{w_{ji}\}_{i=1}^{\infty} \mid \{\sigma_i\}_{i=0}^{\infty} \in Seq(\Sigma, \sigma_0), \sigma_i = (w_{Ei}; w_{1i}, \ldots, w_{ji}, \ldots, w_{ni})\}$.

Now, we associate certain languages with an initial configuration:

Definition 9 *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 $j = 1, 2, \ldots, n$.

Two important selection techniques in dialogue are the turn-taking system and the adjacency pairs. If we want to provide a formal language account of turn-taking, we should focus on the most important traits of this phenomenon, and make it susceptible to formalization. In order to do so, we define different *derivation modes* that control how long an agent can act in the environmental state:

Definition 10 *Let $\Sigma = (E, A_1, ..., A_n)$ be a CGS. And let $w_E = x_1x_2...x_r$ and $w'_E = y_1y_2...y_r$ 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 5. We write that:*

$w_E \overset{a}{\Longrightarrow}{}^{\leq k}_{A_i} w'_E$ *iff* $w_E \overset{a}{\Longrightarrow}{}^{\leq k'}_{A_i} w'_E$*, for some* $k' \leq k$;
$w_E \overset{a}{\Longrightarrow}{}^{\geq k}_{A_i} w'_E$ *iff* $w_E \overset{a}{\Longrightarrow}{}^{\leq k'}_{A_i} w'_E$*, for some* $k' \geq k$;
$w_E \overset{a}{\Longrightarrow}{}^{*}_{A_i} w'_E$ *iff* $w_E \overset{a}{\Longrightarrow}{}^{k}_{A_i} w'_E$*, for some* k;
$w_E \overset{a}{\Longrightarrow}{}^{t}_{A_i} w'_E$ *iff* $w_E \overset{a}{\Longrightarrow}{}^{*}_{A_i} w'_E$ *and there is no* $z \neq y$ *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.

It seems quite common in talk exchanges to find paired actions. Notions such as adjacency pairs, reactive pressures, discourse expectations etc. intend to account for the fact that utterances produced in dialogue are somehow determined and constrained by preceding utterances in the talk exchange. Mapping $\psi_i(w_E, w_i)$ fulfils in CGS a function analogous to the one carried out by all the above notions in their respective conversational models. This mapping establishes which actions are allowed for agent A_i at any given moment.

Closing a dialogue implies that participants stop their conversational activity *because they have reached their goal* in the talk exchange. For deciding when the computation terminates, we have to determine which string is to be considered as the reference point to signal the end of the derivation. We can identify at least three different styles of closing derivation process in CGS:

Definition 11 *Let $\Sigma = (E, A_1, ..., A_n)$ be a CGS as in Definition 1. Derivation in Σ terminates in:*

– *Style (ex) iff for $A_1, ..., A_n$, $\exists A_i : w_i \in T_i$, $1 \leq i \leq n$;*
– *Style (all) iff for $A_1, ..., A_n$, $\forall A_i : w_i \in T_i$, $1 \leq i \leq n$;*
– *Style (one) iff for $A_1, ..., A_n$, $A_i : w_i \in T_i$, $1 \leq i \leq n$.*

According to the above definition, a derivation process ends in style *(ex)* if there is *some* agent A_i that has reached a terminal string. It ends in style *(all)* if *every* agent in the system has a terminal string as state. And it finishes in style *(one)* if there is *one* distinguished agent whose state contains a terminal string. Styles *(all)*, *(ex)* and *(one)* might account for three different ways of closing a dialogue.

The following simple example illustrates how CGS work.

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

- $E = (V_E, P_E)$,
  - $V_E = \{a, x, y\}$;
  - $P_E = \{a \to b^2, b \to a^2, x \to x, y \to y\}$.
- $A_1 = (V_1, P_1, R_1, \varphi_1, \psi_1, \pi_1, \rho_1)$ *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.

4 Final Remarks and Future Work

In this paper we have introduced a formal model of dialogue based on grammar systems. Conversational grammar systems are able to model dialogue 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. CGS present some advantages to account for dialogue: a) generation process is highly *modularised* by a distributed system of contributing agents; b) it is *contextualized*, linguistic agents re-define their capabilities according to context conditions given by mappings; c) and *emergent*, it emerges from current competence of the collection of active agents.

Moreover, we claim that CGS provides a powerful framework for formalizing any type of *inter-action*, both among agents and among agents and the environment. Of course, a topic where context and interaction among agents is essential is the field of dialogue modelling and its applications to the design of effective and user-friendly computer dialogue systems where we think our model can be directly applied.

Finally, it seems this system is quite easy to implement, due to the simplicity of the formalism and the computational background of the multi-agent theory we use. Achieving a valid and simple computational implementation of this formal framework is the major research line for the future. A simple example of implementation of a variant of this model can be found in [4].

References

1. Allwood, J.: Linguistic Communication as Action and Cooperation. Gothenburg Monographs in Linguistics 2, Göteborg University (1976)
2. Allwood, J.: Obligations and Options in Dialogue. Think **3** (1994) 9–18.
3. Austin, J.L.: How to Do Things with Words. Harvard University Press, Cambridge (1962).
4. Bel-Enguix, G., Grando, A., Jiménez-López, M.D.: 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. LNCS 4088, Springer, Berlin (2006) 10–21
5. Bunt, H.C.: Context and Dialogue Control. Think **3** (1994) 19–30.
6. Bunt, H.: Dialogue Pragmatics and Context Specification. In: Bunt, H., Black, W. (eds.): Abduction, Belief and Content in Dialogue. John Benjamins, Amsterdam (2000) 81–150.
7. Clark, H.H.: Using Language. Cambridge University Press, Cambridge (1996).
8. Csuhaj-Varjú, E., Dassow, J., Kelemen, J., Păun, Gh.: Grammar Systems: A Grammatical Approach to Distribution and Cooperation. Gordon and Breach, London (1994)
9. Korta, K., Larrazabal J.: Dialogue as Action. In: Korta, K., Larrazabal J. (eds.): Semantics and Pragmatics of Natural Language: Logical and Computational Aspects. Proceedings of the Donostia-Toulouse 1993 Workshop. ILCLI, San Sebastián (1993) 7–23
10. Levinson, S.C.: Pragmatics. Cambridge University Press, Cambridge (1983).
11. Norman, M., Thomas, P.: The Very Idea. Informing HCI Design from Conversation Analysis. In: Luff, P., Gilbert, N., Frohlich, D. (eds.): Computers and Conversation. Academic Press, London (1990) 51–65.
12. Ogden, W.C., Bernick, Ph.: Using Natural Language Interfaces. In: Helander, M.G., Landauer, T.K., Prabhu, P.V. (eds.): Handbook of Human-Computer Interaction. Elsevier, Amsterdam (1997) 137–161.
13. Păun, Gh. (ed.): Artificial Life: Grammatical Models. Black Sea University Press, Bucharest (1995).
14. Robinson, H.: Towards a Sociology of Human-Computer Interaction. A Software Engineer's Perspective. In: Luff, P., Gilbert, N., Frohlich, D. (eds.): Computers and Conversation. Academic Press, London (1990) 39–49.
15. Rozenberg, G., Salomaa, A.: Handbook of Formal Languages. Springer, Berlin (1997)
16. Searle, J.R.: Speech Acts. An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969).
17. Sharrock, W., Anderson, B.: Epilogue: The Definition of Alternatives: Some Sources of Confusion in Interdisciplinary Discussion. In: Button, G., Lee, J.R.E. (eds.): Talk and Social Organization. Multilingual Matters LTD, Clevedon (1987) 290–321.