Interacting with Computers 24 (2012) 2, pp. 78-90
DOI: 10.1016/j.intcom.2012.02.003

Interpretation and generation incremental management in natural interaction systems ☆

David del Valle-Agudo *, Javier Calle-Gómez, Dolores Cuadra-Fernández, Jessica Rivero-Espinosa

Department of Computer Science and Engineering, Carlos III University of Madrid, Av. Universidad 30, 28911 Leganés, Madrid, Spain

Abstract: Human interaction develops as an exchange of contributions between participants. The construction of a contribution is not an activity unilaterally created by the participant who produces it, but rather it con-stitutes a combined activity between the producer and the rest of the participants who take part in the interaction, by means of simultaneous feedback. This paper presents an incremental approach (without losing sight of how turns are produced throughout time), in which the interpretation of contributions is done as they take place, and the final generated contributions are the result of constant rectifications, reformulations and cancellations of the initially formulated contributions. The Continuity Manager and the Processes Coordinator components are proposed. The integration of these components in natural interaction systems allow for a joint approach to these problems. Both have been implemented and eval-uated in a real framework called LaBDA-Interactor System which has been applied to the ''dictation domain''. We found that the degree of naturalness of this turn-taking approach is very close to the human one and it significantly improves the interaction cycle.

Keywords: Human computer interaction, Natural interaction, Presentation manager, Incremental interpretation, Incremental generation, Continuity management.

1. Introduction

This paper is about human–computer interaction, and more specifically, about natural – or human-like – interaction systems, which try to make technology accessible through the same codes, modalities and procedures that people use when interacting with one another, in other words, when developing a natural interaction.

Natural interaction is divided into three different levels: local (how each of the goals progress internally), global (the existing relationships between different goals) and temporal (the way in which participant turns are coordinated and how and when they are produced throughout time). Over the last few years, there have been several important advancements in the development of this type of technologies, especially those related to local and global interaction organization. Temporal interaction organization has barely been dealt with up to now.

This temporal interaction organization refers to the strategy by which participants carry out the turn-taking in the interaction. This means the way in which they decide when to participate in the interaction and with what kind of contributions. Generally, interaction systems tend to simplify turn-taking, considering it as a ''pass the baton'' process by which the floor (the turn of speech or right to speak) is passed from one participant to another in an organized manner, in which only the person in possession of the floor can contribute to the interaction and can unilaterally determine how to proceed. Under this turn-taking model, called interaction cycle (Vanderheiden and Zimmermann, 2005), the validity of a contribution is maintained from the moment that it is formulated through its complete expression within a turn, and it does not require signaling the participant's intention of acquiring, maintaining or releasing the attention of his interlocutors.

For certain interaction domains, for example transactional or information recuperation domains, this could be a valid approach. However, as interaction systems' interactive abilities improve (especially their proactive capabilities and representation of the changing circumstances around the interaction) this interactive behavior becomes mechanical and unnatural, and the application of turn-taking mechanisms that better reproduce what happens during human interaction becomes essential.

Human interaction is, in fact, a joint activity (Sacks et al., 1974) in which participants require evidences that they have succeeded in performing their actions (Calle et al., 2004). To this end, participants offer public displays of acceptance or rejection of such actions. These public displays are shown simultaneously to the production of these actions (primary track) through collateral flow of action (secondary track). Consequently, in a natural interaction situation, the participants produce their contributions during an incremental generation process (Kilger and Finkler, 1995) through which the participants, before beginning their turn, only have one

* Corresponding author.
E-mail addresses: dvalle@inf.uc3m.es (D. del Valle-Agudo), fcalle@inf.uc3m.es (J. Calle-Gómez), dcuadra@inf.uc3m.es (D. Cuadra-Fernández), jrivero@inf.uc3m.es (J. Rivero-Espinosa).

formulation prior to the one they are about to develop, and it is while they are developing it that they refine the content and form thanks to the simultaneous feedback that they receive from their interlocutors and with the successive changes produced in the state of interaction and in the circumstances around it (the context related to the session, users, situation and emotions). Additionally, the temporary development of the turns offers information that is very important for the correct modeling of the participant state of turns and the candidates to take the floor. This information is required to apply turn management strategies (Stivers et al., 2009).

To support the temporal development of natural interaction, incremental treatment of interpretation and generation processes are required, as well as the detection and synthesis of floor management markers that are produced as alterations of the temporary continuity of the development of the turns (momentary delay, reinitiation of turns, pauses and continuation, among others (Jefferson, 1989). Throughout this paper we will describe a proposal to develop this type of presentation strategies, formed by the Continuity Manager and Processes Coordinator components. This paper starts with a brief review of the current state of the art and describes the components proposed within the LaBDA-Interactor framework. Finally, we have included the methodology used for our evaluation and some of our conclusions.

2. Related work

Temporal development of human-like interaction should focus on discursive, multimodal, mixed-initiative and joint-action systems, respectively. Discursive systems are required as the manner and moment in which turn-taking during the interaction is derived from its profound comprehension and the surrounding circumstances. In Multimodal systems, natural interaction turn-taking is largely managed by paralanguage. A mixed-initiative system is also required because, in natural interaction, the initiative to add, eliminate or progress goals can arise at any moment and come from any of the participants. And, finally, solutions based on joint-action are required because, in the last instance, equilibrium between the participants' goals and their compromises determine how the turns in the interaction are developed, rectified, reformulated or interrupted.

Among the proposals that deal with the resolution of some of the incremental processing issues, barge-in systems (Komatani and Rudnicky, 2009; Komatani et al., 2008) are particularly remarkable. These systems are capable of managing the interrup-tion of its own contributions when the user starts a new simulta-neous utterance. Nevertheless, these systems do not consider the possibility of reformulating its contributions as consequence of these simultaneous user utterances (or as a consequence of changes that take place in interaction circumstances). In Ymir (Tur-unen and Hakulinen, 2000), FADE (Reithinger and Kipp, 1998) and VM-GEN (Kilger and Finkler, 1995) user utterance interpretation and system utterance generation processes are independently ad-dressed. FADE and VM-GEN also contemplate an incremental development of the interpretation, allowing for the dynamic updating of the context as the contribution is being expressed, and not only after its complete interpretation. Along with all of this, VM-GEN is capable of supporting incremental generation, making the rectification and reformulation of the system's contri-butions as the interaction context changes possible.

Despite the virtues of these systems, none of them are capable of interpreting or generating the attention-management markers that participants include in their utterances as an alteration in their continuity. These markers are expressed in the form of disfluencies (Bunt, 2009) (hesitations, discontinuities, repetitions, silences, filled silences, etc.) when participants perform certain types of presentation strategies (commit-and-repeat (Goodwing, 1981) and commit-and-repair (Jefferson, 1989), among others) with the aim of requesting, maintaining or releasing the interlocutors' attention. Furthermore, they cannot monitor the state of activity of the participants' turns in order to provide support for estimating who are the speaker and the candidates to take the floor and for performing the system's turn-taking decisions.

In conclusion, although there are solutions that go beyond the development of the sequence of turns that constitute the interaction cycle, none of them completely addresses the incremental management of the interpretation and generation processes, with continuity management and the ability to support advanced turn-taking management. The omission of these aspects in hu-man–computer interaction makes turn-taking a mechanical and artificial process. The generation of more natural results means redesigning the system's architecture applied to the development of natural interaction systems, as well as some of the processes that they develop.

3. Domain of interaction

We searched for interaction domains in which advanced turn-taking skills would be necessary and those in which the influence of the problems related to other levels of natural interaction were minimized (voice recognition and synthesis, natural language processing, multimodal adaptation, etc.).

In choosing an appropriate interaction domain several domains proposed within the framework of the national competitive project SemAnts [TSI-020100-2009-419] related to the development of games and other collaborative tasks were analyzed. From each of these domains we compiled a preliminary corpus, applying the person-person technique (two people interacting in the scenario of the proposed domain, one in the system's role and the other as the user). During the acquisition phase, the participants were not instructed on the specific script to be developed, or on the set of expressions, modalities or turn-taking rules that they should restrict their interactions to.

After analyzing the acquired interactions, the selected domain was the ''dictation domain''. In this domain the system played the role of a boss (the participant who dictates a text), and users the role of secretary (the participant who copies the text dictated by the boss). In this domain the user's inputs are multimodal, because they come from the user's speech and gestures and from the keyboard. In the same way, system's outputs are multimodal and they are composed of both speech and gestures.

The dictation domain was very suitable for evaluating advanced turn-taking. It includes primary and secondary contributions and long contributions subjected to reformulations, overlaps, interruptions and changes of speaker. In this corpus participants carry out goals for: dictating the text to the user; notifying orthographic and typographic errors in the copied text; asking the system to indicate orthographic points; requesting the system to repeat parts of the text, to pause or continue the dictation or to spell a word; and some other goals to reinforce the user's commitment when it decreases (see Table 1).

4. Cognitive architecture for advanced turn-taking

This section describes the LaBDA-Interactor System's architecture as a framework for our proposal. This architecture is based on a joint-action dialog model (i.e., the threads model (Bunt, 2009) and implemented on a blackboard-oriented, multi-agent platform (Ecosystem) Roberts and Bavelas, 1996. Its knowledge models are implemented as one or more agents (depending on the model), running on a standalone machine or distributed

Table 1
Goals carried out in the dictation domain.

Goals	Description
Dictate	The system dictates the copied text to the user
Correct error	The system corrects a spelling or typographical error in the text that has been copied by the user
Orthography	This is opened by the user to check orthography of the dictated text
Repeat	The user requests repetition of part of the dictation
Recapitulate	Reinforcement technique
Spelling out	The system spells out a word (on its own initiative or when requested by the user)
Wait/continue	The user asks for a pause in the dictation, which will then be resumed
Apologize	The user apologizes for errors

through a LAN. Agents offer services to each other and results usually come from the overall collaboration. The influence of the different knowledge models in the interaction is simultaneous, that is, the system's interactive behavior is the result of its autonomous processing and co-operation.

The interaction developed between people, i.e. *human interaction*, requires using a large amount of knowledge and numerous skills. The proposed architecture gives structure to the skills and knowledge in the following group of components (Fig. 1): Interface Components, Ontology, Interaction Manager and Presentation Manager.

Firstly, it requires the ability to acquire the expressions naturally produced by users through these different modalities, describing them by means of semantic structures that the system can understand. Similarly, the ability to synthesize the system's contributions, initially represented by information flows, into natural expressions using the appropriate modalities is required. With the purpose of providing such services the architecture includes *Interface Components*, having speech and gesture recognizers, such as graphical interfaces, speech synthesizers and natural language processors for different modalities applied to interaction. Speech and gesture recognizers, graphical interfaces and speech synthesizers compose the physical layer of the Interface Components. These acquire the users' expressions in different modalities (voice, gestures, etc.). Likewise, the logical layer of the Interface Components is formed by natural language processors, whose function is to normalize the expressions acquired from the user, following a more formal and structured notation, striving to preserve the most of the semantic contents possible, doing the opposite in the case of expressions generated by the system. Thus, expressions as "Hello" or "Good morning", or even saluting with a gesture, can be coded as a common expression "greeting". Within the dictation domain (Exa. 1), the expression "'Escondido' la primera con mayúscula" ("'Hidden' with capital 'H'") can be simplified as "inform(matter = "orthography", subject = "uppercase", object = "Hidden")".

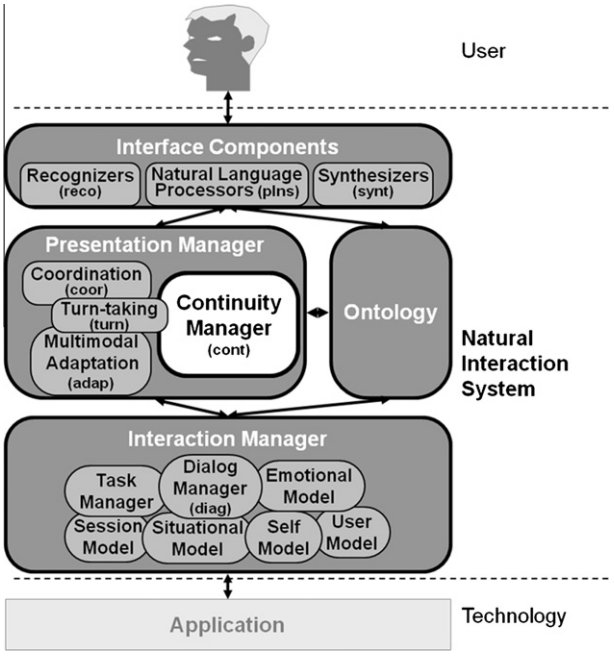

Fig. 1. Cognitive architecture for a natural interaction system.

Given that the signifiers expressed by the users do not have fixed equivalences as regards to the concepts they refer to (due to matters as the origin of the speakers, their education and their perception of the world), it becomes necessary to include in the architecture a component that makes comparable the concepts managed by the system and those managed by the user. Likewise, the different system components may use different symbolic representations for the same concept; therefore, it is also necessary to collect all different representations to make all the components of the system compatible. Furthermore, two completely different concepts can share a certain semantic relationship, thus making them comparable. To solve such problems, an *Ontology* component has been included in the architecture.

The *Interaction Manager* contains the components responsible for the system's interactive behavior. Its core is the *Dialog Manager*, which represents the participants' shared goals (to complete the dictation, to correct spelling mistakes, etc.), as well as their own discursive and individual goals (to dictate or to copy the text, to notify or update spelling mistakes). In this paper we have applied a specific dialog model: the threads model (Bunt, 2009). This is a symbolic (non-statistical) intentional joint-action dialog model which is based on the notion of dialog threads. Threads are acquired as sub-dialog abstractions from the corpus, and instanced during the interaction with a set of features fixing state, context, attention, etc., of the interaction. Once a thread instance has been set, it can be developed, canceled or temporarily abandoned to be reopened later. The Dialog Manager also represents, in a hierarchical structure, the global connections among the different thread instances (intentional structure) and the order of development of these instances in the interaction (focus). For a thread instance to be progressed, it has to be well committed to by the participants.

On the other hand, joint threads progress as a result of the need for a participant to develop his/her own goals. The system urgency to reach a goal is given by a criticality function, an expression dependent on variables of the situation, session, state of threads, etc. The role of criticality is determined by the component that inserts the goal and is described as the measure of the need to accomplish a goal, having a continuous value ranging from 0 to 1 (0 being a completely expendable goal, and 1 would be attributed to goal that, if they are not met, would cause the dialog to fail). This function can be variable-dependant as, for example, the time lapsed from its insertion, or from the last time this goal was

highlighted, the value of certain lines of context, the commitment status of the thread or other threads, etc. These variables change either as consequence of intervention interpretation and generation, or as result of changes in the circumstances around the interaction (the context related to the session, users, situation and emotions). The criticality value of a system's goal could reach thresholds that trigger new generative processes.

Within the Interaction Manager there is also a component that makes technology accessible for users, the *Task Model*, which represents the operative layer of natural interaction systems and its back-end. The development of interaction implies achieving states in which the system recovers information from external services (for example, to solve user queries) or triggering effects on the technology it operates (execution of transactions and commands, programming events, etc.). This set of skills varies depending on the domain of interaction. For example, in the dictation domain, a file must be opened, containing the text to be dictated, organized inputs and feedback of its dialog outputs. The system component responsible for interacting with external services must be easily adaptable for different applications, in other words, it must allow for an easy addition of new skills whenever the domain of application so requires. The Task Model is comprised by a *reasoning engine* (which applies logical algorithms and conditions to the resolution of the tasks to be solved); and a *persistence layer* (granting access to applications and enabling their control).

The Interaction Manager also contains other knowledge models to represent additional sociolinguistic knowledge of influence on the interaction:

- The *Session Model* keeps the interaction context and log, and solves anaphoric and deictic references.
- The *User Model* performs the interlocutor characterization, predicting unknown features to adapt interaction to the current user, supported either by ad hoc built stereotypes or by past use; for example, it can decide the proper rate and rhythm of dictation for current users, based on the behavior of similar former users.
- The *Emotional Model* manages the emotions influencing the conversation; deciding on the convenience of interrupting the speaker based on his or her emotional state; and in the case of doing so, the need to introduce techniques to mitigate the emotional impact of this action may be assessed.
- The *Situation Model* deals with the circumstantial aspects surrounding the interaction; by characterizing the situation, it enables the binding of the relevant knowledge to be applied anytime within every model, enhancing efficiency and eventually avoiding ambiguity (for example, the utterance "spell" only has one interpretation in the dictation domain).
- The *Self Model con*tains the system's own goals and beliefs; for example, warning the user (of a problem, a timer, etc.) can be as urgent as interrupting the current activity (dictation).

In this architecture, the Presentation Manager is the component performing the multimodal adaptation of utterances (López-Cózar and Spoken, 2005). In this p*roposal, i*t is also responsible for some new functions in order to manage the incremental processing of the interaction and its temporal organizational level. These new functions are: Turn-taking Management, Continuity Management and Processes Coordination. Turn-taking Management enables the system to make conjectures about the state of the turns that participants yield, as well as more complex turn-taking requirements. It is important to consider issues as the track of each turn execution (if it is a primary or secondary contribution), the state of the turns of speech and the participants' position regarding future turns (if the participant is awaiting his turn, if he has been appointed as next speaker, etc.). This information makes estimation possible when required to produce new system interventions according to the rules of human interactions (Vanderheiden and Zimmermann, 2005). The Turn-taking Manager is the component in charge of these new functions.

On the other hand, Continuity Management and Processes Coordination are the Presentation Manager functions that are responsible for the incremental processing of the interaction. Continuity Management entails checking units of expression with complete interpretative meanings (dialog acts Besser and Alexandersson, 2008) during the user's contributions, and managing any possible eventualities (rectification, reformulation and self-interruptions) that can occur during the system's contributions. Finally, the independence of interpretation and generation processes requires access monitoring and coordination of the resources shared by both processes (status of joint goals, users, sessions, situations and knowledge of emotions, the system's self-knowledge, etc.). This function is called Processes Coordination. Both functions are the core of this project and they will be further explained throughout the following sections.

5. Continuity management

The Continuity Manager – in collaboration with the Processes Coordinator, the Turn-taking Manager and the Dialog Manager – addresses the incremental development of the interpretation and generation processes of contributions in natural interaction.

From an incremental point of view, interpretation and generation are processes with a continuous development over time that require real-time execution by the participants. However, these processes (that occur parallel in time) use shared resources (the knowledge relative to the state of interaction, session, situation, users' auto-modeling and emotions) and, as a consequence, access to such resources must be controlled. To comply with the real-time restrictions, access control cannot be simply limited to blocking these processes while another interpretation or generation process is being developed. It should be noted that the time required to interpret or generate a complete contribution is usually longer than the maximum delay within which the interlocutors expect to receive public displays of acceptation or rejection of their actions (each individual action composing their contributions). Consequently, such processes must be divided into smaller units, in order to allow the interpretation of these individual actions and the generation of their public displays within the maximum expected delay. Fig. 2 shows the incremental interpretation and generation processes executed during a portion of the interaction collected in Ex. 1.

The duration of these delays depends on circumstantial and socio-cultural issues. Though, some linguistic and sociolinguistic works (Thórisson et al., 2002) addressing the characterization of the gaps between participants' turns, have delimited this time between 100 and 500 ms. The time interval that participants are willing to wait until the reception of the next turn is related to the time they are willing to wait until the reception of the public displays to their complete turns. Standards on signal processing (International Telecommunication Union (ITU-T), 2003) apply similar values (between 150 and 400 ms.). Working from this assumption, the manner in which the Continuity Manager develops the aforementioned processes is described below.

5.1. Incremental interpretation

User contributions, rather than being interpreted in a single process upon their complete acquisition, are interpreted throughout the process through a series of partial subprocesses. The users' expressions are acquired little by little through the Interface

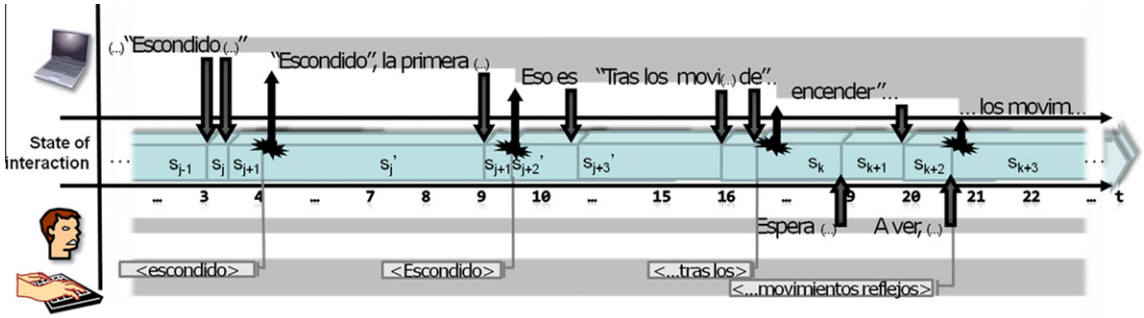

Fig. 2. Incremental interpretation and generation processes for an excerpt of the interaction collected in Ex. 1.

Components. As the Continuity Manager receives the contribution fragments from the Interface Components, it combines them with those previously received through the same modality in larger units. These units are sent to that Natural Language Processor which is linked to this modality, and this processor may detect portions of contribution with possible meanings in the interaction domain. Continuity Manager sends to the Multimodal Adaptor the possible meanings detected in all the modalities in order to perform the fusion of these portions of contributions in complete communicative actions. When this happens, the contribution portion received is provisionally considered a complete interlocutor contribution and the Dialog Manager (via the Processes Coordinator) will be requested the interpretation of the dialog acts, generating opportune developments in the interaction state and in the associated knowledge about the session context, users, situation and emotions.

Suppose that, within the dictation domain, when the user reaches the word "movimientos" ("movements"), the user asks the system: "¿Qué has dicho antes de cigarrillo?" ("What did you say before cigarette?"). Fig. 3 shows how, under this conceptualization, the interpretation of the users' contribution is developed. Among the Interface Physical Components are the input recognizers (labeled as "*reco*" in the figure) that acquires the users' speech, gestures and the copied text. Throughout the development of the users' contributions, the system's recognizers send to the Continuity Manager (labeled as "*cont*") the partial fragments of the users' contributions in real time (at callings 1, 7, 13, 23, 29 and 35). The Continuity Manager identifies those cases of disfluencies that it may contain (silences, throat clearing, repetition of the last words spoken or any other type of filled silences, among others) and interprets their meaning as turn-taking markers. As a result, we obtain some cases of turn-taking markers (beginning of activity, commit-and-repeat, commit-and-repair, turn requests and assignments, etc.) and disfluency-free portions. Once the turn-taking markers have been identified the Turn-taking Status is updated (at callings 2, 8, 14, 24, 30 and 36) in the Turn-taking Manager (labeled as "*turn*"). This is a key process in the establishment of speculation regarding the state of activity of the participants' turns, who is the speaker and who are the candidates to speak afterwards.

Meanwhile, the Continuity Manager sends the disfluency-free portions to the Natural Language Processors and Multimodal Adaptor ("*plns*" and "*adap*", respectively) in order to detect complete dialog acts (at callings 3–6, 9–12, 25–18, 25–28, 31–34 and 37–40). These portions are subjected to incremental processes of natural-language interpretation (resolved in the system's Natural Language Processors) and multimodal fusion (performed by the Multimodal Adaptor). Applied grammatical formalisms and the group of valid domain expressions are determined by the system's Natural Language Processors. In this approach they operate by simple context-free grammars.

As the token sequence increases, the natural language interpretation of the contribution is revised iteratively, with the goal of identifying expression fragments with complete interactive meaning. The resulting dialog acts will be sent (see callings 19 and 20) to the Dialog Manager ("*dial*") through the Processes Coordinator ("*coor*"). The Dialog Manager carries out the dialog interpretation of such dialog acts and updates its interaction state as well as the knowledge stored in the rest of components of the Interaction Manager (Session, Emotional, Situational, User and Self Models, labeled "*know*").

Thus, at calling number 23 the system has acquired and interpreted the user's portion of contribution "¿qué has dicho?" ("What did you say?"), which has a complete interactive meaning (the user needs the text to be repeated from the word "movements" onwards).

Sometimes these partial interpretations could be imprecise or wrong and, as the contribution goes on, the system must refine his provisional interpretation. In the example, after this preliminary interpretation the user says "...antes de cigarrillo" ("...before cigarette") between callings 23 and 44. This means that the user really needs the text to be repeated from the word "cigarette" (instead of "movements"). Therefore, an incremental focus of the interpretation allows for iterative refining of the dialog acts, which are obtained as the new contribution fragments are received. In some cases, this results in new dialog acts with which to update the interaction state and the rest of components of the Interaction Manager. In others, called reinterpretations, there is a rectification of some of dialog acts previously interpreted in more precise cases. In these cases of reinterpretation, the interpretation of the new dialog acts should be preceded by a restoration of the interaction knowledge to the state prior to the interpretation of the obsolete dialog acts. This restoration could even undo the changes made by other interpretation and generation processes, some of which might have led to the development of goals, which will now be canceled. In these cases it is possible to develop clarifications, apologies or rectifications by inserting new discursive goals in the Dialog Manager. With this, the system will identify that the user actually needs to know what words came before "cigarette", thus discarding the initial interpretation.

This way, the updating of the interaction state and the rest of components of the Interaction Manager (through the detection of the new dialog acts) and the updating of the state of the participants' turns, the floor and candidates to use their turns of speech (through the detection of the cessation or continuation of the activity and turn-taking markers) are produced during the development of the contribution (and not only after its complete acquisition), simultaneously affecting the system's current contribution course by canceling it or leading to the generation of a new one. These tasks are conducted, respectively, by the Dialog Manager and Turn-taking Manager.

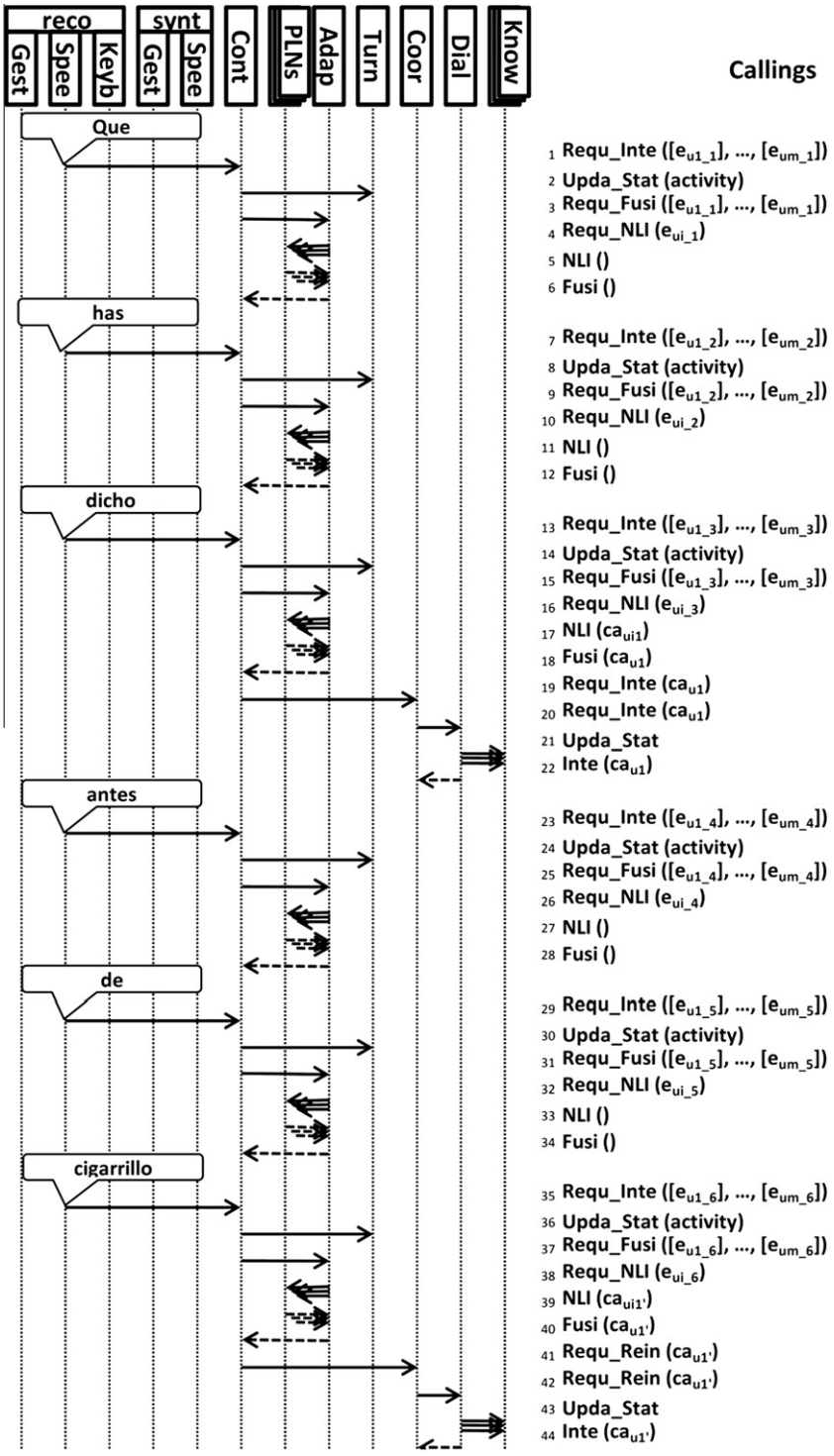

Fig. 3. Incremental interpretation of the "What did you say before cigarette?" contribution.

5.2. Incremental generation

As an effect of the changes to the circumstances, due to the contributions produced simultaneously by the rest of the participants or by the detection of interaction errors, the contribution that the system develops in its current turn could be rendered obsolete, making its rectification, reformulation or interruption necessary. Even though the reformulation of a contribution is developed by the Dialog Manager and the turn-taking decision comes from the Turn-taking Manager, the task of coordinating the incremental generation process, adapting the new fragments of the system's contribution to those already expressed (even with rectification or self-interruption situations) is developed by the Continuity Manager.

To manage the continuity of the reformulation with the greatest fluidity possible, the system's contribution generation process must be split into two different subprocesses: (a) the formulation of a contribution (labeled as "*Form*") and; (b) the confirmation of its expressions synthesis (labeled as "Conf"). Formulation is the first generation phase, and it is produced in the Dialog Manager (under the control of the Processes Coordinator) previously to the begin-

ning of its synthesis in the channel. At this moment, the system's contribution is completely built. The Continuity Manager incorporates this contribution into the system's expression flow (replacing the previously formulated contribution or combining both as smoothly as possible, in each case). Afterwards, when the contribution has been formulated and processed by the Multimodal Adaptor, and when the Interface Components have started to express it through the channel, the system requires monitoring which part of its contribution has been expressed up to the moment (and what is left). This is done during the confirmations of the synthesis of the expression. This way, the system will be able to manage possible future reformulations of its contribution. The processes of confirmation of expression synthesis consist in notifying the Dialog Manager when fragments with a complete interactive significance have been expressed, in order to update the interaction state.

This proposal considers, with regard to dialog management, that the atomic units in which the system structures the formulation of its contribution are the dialog acts and, therefore, the complete expression of each of these dialog acts will lead to a new synthesis confirmation of expressions in the Dialog Manager. With this mechanism, the Continuity Manager can discover what portion of the contribution the system has expressed during its turn up to that point, as well as what part it still has to express. Identifying the portion of the contribution that has been expressed up to the moment is fundamental to the subsequent management of the reformulation. Fig. 4 shows the incremental generation of the system's utterance ''Escondido tras los movimientos reflejos de encender un cigarrillo'' (''Hidden behind the reflex movements of lighting a cigarette'').

Reformulation occurs when the system has a contribution in course and a new turn-taking decision is produced that results in a new formulation (for example, ''Escondido tras los|... Escondido, la primera con mayúscula'', ''Hidden behind the|... Hidden with a cap-ital H'' Fig. 5). In these cases, it is considered that a transition point exists between the formulations if the new one begins with a com-patible dialog act with one of the dialog acts from the previous for-mulation. Depending on whether a transition point between the contributions (marked as ''|'') exists or not and where it is found, *soft reformulation*, *rectification* and *auto-interruption* cases can be distin-guished. Soft reformulation is produced when both formulations, obsolete and new, fit and the transition point is subsequent to the position expressed up to that moment. Rectification occurs when both formulations fit, but the transition point has already been ex-pressed. When the formulations don't fit, the result is auto-interruption.

Finally, among the Continuity Manager's functions is that of notifying the Turn Manager when there is activity in the system's turn or when it ceases (developed during the confirmation processes). The synthesizing function is also present, by means of alternations in the temporary continuity of the system's turn, some of the turn management markers (momentary turn delay, stop and continuation, etc.) when it is defined as such in the formulation received from the Dialog Manager.

Despite all of this, the generation process is not restricted to the expression of what was initially formulated. The system can update its contribution as it goes in order to adjust it to the changes produced simultaneously in the interaction state, in the changing circumstances around the interaction, and in the state of the participant turns (who has taken the floor, the candidates to make use of their turns of speech, etc.).

6. Processes coordination

During the development of the natural interaction, there are several processes that are simultaneously developed in the system. On the one hand, the interpretation of the contributions of the interlocutors, which are continuously generated. On the other hand, the generation of the contributions that the system can develop in parallel with these interpretation processes. Finally, the circumstances that surround the interaction change dynamically and the changes that are produced must be monitored in order to detect when new system generation processes must be set into motion.

Some situations in which several processes occur are represented in Ex. 1. There are simultaneous processes when, for example, one of the participants offers feedback at the same time as the speaker ($t = 28$) or when there are requests to take the floor ($t = 16$), possibly interrupting each other ($t = 18$). Additionally, different events could require the review of the system's turn (taking or canceling the turn or reformulating the current contribution). They could be originated from changes in the circumstances around the interaction ($t = 5$, $t = 9$ and $t = 34$), in the state of interaction ($t = 2$ and $t = 11$) or in the system's proactivity ($t = 22$ and $t = 32$). Other conditions with simultaneous processing occur when the participants overlap one another, either due to an error in estimating who is the current speaker or due to the competition for taking the floor.

In Ex. 2, we observe the loss of naturalness that comes with limiting the system's capacity to develop such processes in parallel, making the development of process coordination strategies that attend to them concurrently a necessity.

With the incremental treatment of contribution interpretation and generation, these continuous processes are broken down into discrete subprocesses of negligible execution time when compared to the production time of natural language: interpretation of dialog acts, formulation of contributions and synthesis confirmation of expressions (Sections 5.1 and 5.2). Given that any of these subprocesses requires access to a series of shared resources (knowledge of the interaction state, session, user, emotions, situation and automodel) and that the order in which they are executed is key to determine the final state reached, the continuity management should be complemented with a processes coordination strategy.

For this reason, the approach includes a Processes Coordinator component that bases its strategy on a line, with priority given to the one on which different processes deposit their requests and from which they are extracted in order, one by one. The order established by the requests line is based on the need for assurance that estimations of the knowledge state that reached the system and the interlocutors are similar. In accordance with these criteria, the requests will be interpreted in the following order:

(i) Synthesis confirmation of expressions.
(ii) Interpretation of the interlocutors' new dialog acts.
(iii) Formulation of new system contributions.

The synthesis confirmations are given maximum priority as the system cannot ignore that which it has expressed. It is also related to the interaction state in which it was formulated, losing validity after the interpretation of other dialog acts or the generation of new contributions. The interpretation of the interlocutors' dialog acts should be executed, in the same way, previous to the new progresses that would lead to the generation of new system contributions. Finally, and once all the rest of pending requests have been resolved (from any of the participants), the Processes Coordinator will attend to the formulation of new contributions.

Together with all of this, the execution of dialog acts interpretation requests and the formulation of new contributions should be preceded by the cancelation of the system's contribution in course (when this is the case). In the event of the changes that would lead to the execution of the aforementioned subprocesses in the interaction state, the validity of the pending contribution portion cannot be guaranteed and, in the event that finally the reasons that

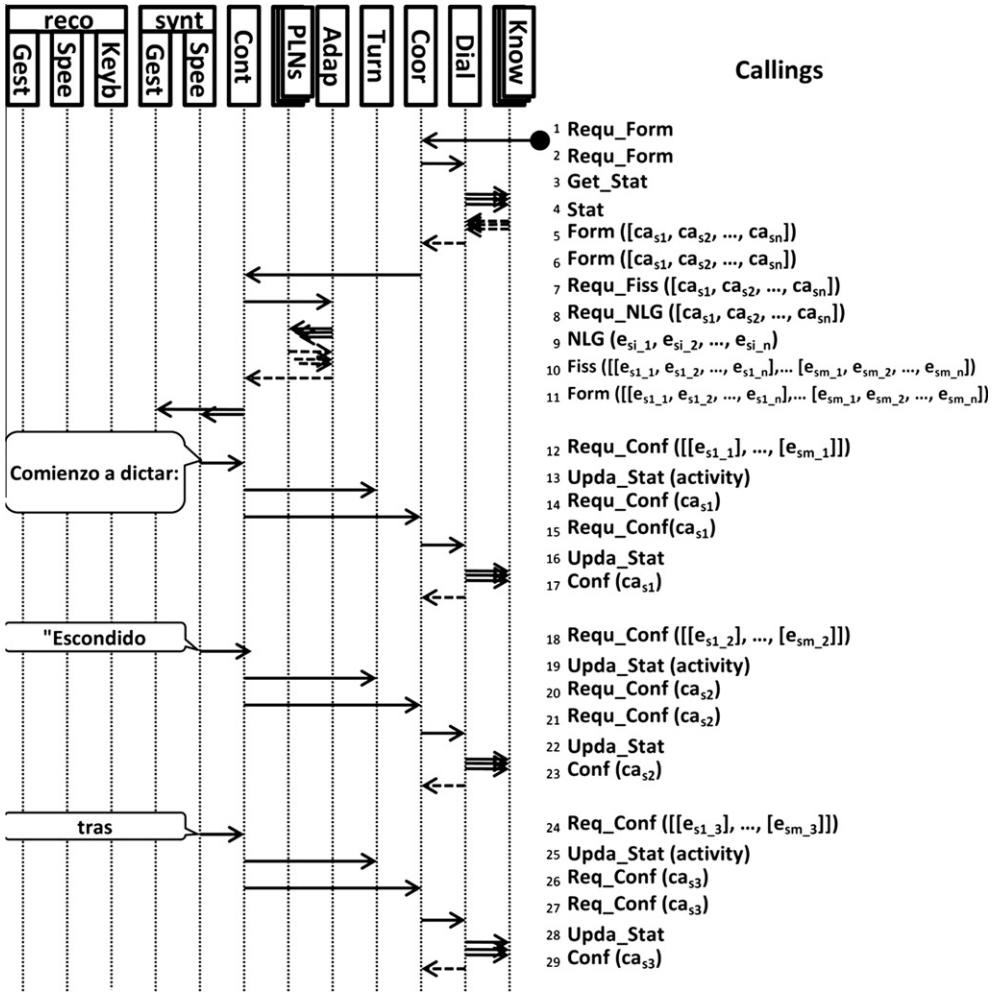

Fig. 4. Incremental generation of the contribution "Begins to dictate: Hidden behind…".

led the system to formulate the contribution that was canceled are still valid, the system will formulate its contribution again. In these cases, the result could be a continuation of the contribution (if there are no changes between the obsolete formulation and the new one) or a soft reformulation, rectification or auto-interruption (if appropriate, based on the degree of said changes).

7. Evaluation settings

This section describes an empirical evaluation of the models described in Sections 5 and 6. These models have been included in a real framework called LaBDA-Interactor System, which consists of a set of Java packages that implement its different models encapsulated in agents. Such agents are distributed across a LAN and its communication is performed through a blackboard supported by a DB and embedded Java supported by the DBMS Oracle 10 g.

This evaluation is based on the works realized by Dybkjær et al. (2004) and other authors like Walker et al. (1998). In order to evaluate this approach, real interactions were took place between the described system and test users. These interactions were restricted to the dictation domain and they took place under different configurations of the interpretation and generation processing.

7.1. Participants

Thirty-nine subjects (28 men, 11 women) participated in the study on September 2010 in the LaBDA Group Laboratories, at the Carlos III University of Madrid. All subjects where native speakers of Spanish, 35 Castilian Spanish, 3 de Mexican Spanish and 1 Colombian Spanish. Their ages (Table 2) ranged from 14 to 62 years (mean: 29.98, standard deviation: 11.57). Of these, 33 used computers frequently and 6 used them occasionally or never. Five participants had not graduated from school, 2 had finished middle school, 2 had finished high school, 12 had superior or university studies, 11 had undergone post-graduate studies, 6 said to be experts in information technologies and 1 said to be expert in Human–Computer Interaction. All subjects were recruited through announcements in the university's bulletin boards, which were directed to students, services personnel, professors and visitors. Of all possible candidates, we made a selection to obtain a wide range of ages and educational backgrounds.

7.2. Design of experiments

Both participants, the user and the system, had the common goal of completing the dictation, taking care of aspects like the orthography, punctuations marks and capitalization. The system and the users did not have the ability to solve the task by themselves (the system does not have the ability of copying the text, while the users do not know the text they have to copy). During the realization of dictations, the system can suspend the dictation to help the users correct their errors and it can also adapt the dictating speed to that of the users. As far as the users are concerned, they are expected to behave as humans: asking the system about spelling, asking it to wait, continue, repeat, etc.

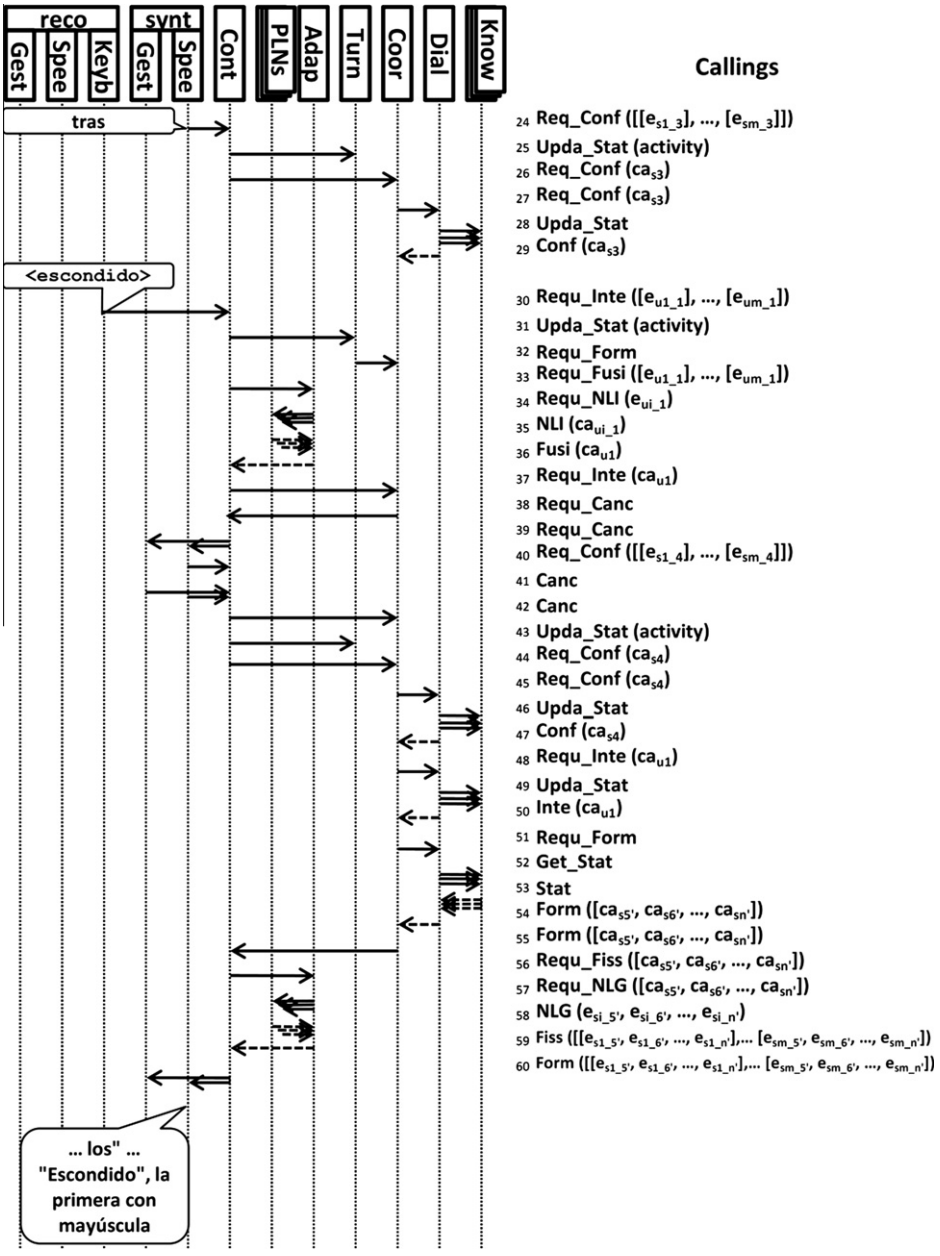

Fig. 5. Reformulation of the contribution "Hidden behind the"..."Hidden" with capital "H".

Each test subject did three different dictations with the system, each one with a different processing configuration: (A) interaction cycle processing (the processes are developed sequentially and not incrementally); (B) incremental processing (the system develops the strategies proposed in this work); (C) Wizard of Oz (a human determines what the system has to say and how it has to behave at all times). These configurations were presented to the user in a random order, so they were unaware of what configuration was being applied. Furthermore, they were also unaware of which configuration is under evaluation (if any), and the parameters considered in the evaluation. However, they were previously trained in the task of copying texts in the same settings in which the experiments were conducted, so they could successfully develop this task afterwards. That training consisted of a brief explanation of the task and tools and a practical exercise. The practical exercise involved the realization of some dictations, which were directed by a human playing the role of a human. As many dictations as the user needed to be familiar with the task and tools were realized. Finally, texts used in dictations (both for practical exercises and for every one of the possible system configurations) were taken from the same set of texts. All these texts had similar difficulty and length, and they were randomly selected from this set following a uniform distribution.

In order to avoid the test subject becoming aware of when the configuration is a machine or human, there are two more participants in the experiments: an experimenter and a typist, who act as interfaces (output and input respectively) between system and user. This way, the test subject and the test experimenter are seated face-to-face, each one with a computer (first one uses it to copy the text, and the other to interact with the system). When the user interacted with a machine configuration (A or B), the typist simulates speech acquisition by typing user utterances. The experimenter read the system's contribution using the speaking rate and pauses specified by the system. When the user interacted

Table 2
Participants' characterization.

	Men	Women
Sex	**28**	**11**

	<25	25–34	35–44	⩾45
Age	**9**	**16**	**8**	**6**

	Did not finish school	School graduate	High school graduate	University studies graduate	Post-graduate	ICT expert	HCI expert
Education	**5**	**2**	**2**	**12**	**11**	**6**	**1**

	Occasionally or never	Usually
Use of PC	6	33

with the human configuration (C), the experimenter watched in the screen the text to be dictated, the text the user had copied and the typed user's utterances. In this case, the experimenter was the participant who developed the interaction (instead of the system). The experimenter acts the same way in all three configurations, so the user cannot find differences between them in the interfaces. The experimenter was previously trained in his tasks and both participants, experimenter and typist, were completely impartial in the experiments.

7.3. Metrics

The measurement of results was based on some objective (technical evaluation) and subjective parameters (usability evaluation). The set of technical parameters were taken from previous studies by other authors (Dybkjær et al., 2004; Ward and Pellom, 2003). We considered the duration and number of exchanged words in the interaction and the number of contributions produced.

In order to evaluate the usability, we required the users to assess their satisfaction with each configuration from 0 to 10 in a Likert scale (Fig. 6). Moreover, they were asked to fill out a questionnaire to assess the naturalness of the interaction for these configurations (Fig. 7). The form contained questions in order to know which configuration produced more (or less) dynamism (questions 6 and 5), organization (8 and 7), co-operativity (9 and 10), were more suitable for the task (11 and 12), the convenience of the interaction (13 and 14) and if the user preferred (or rejected) any of them (15 and 16). Each category was marked with 1 point, for the best configuration (if any of them looked better than the rest), or 0 the worst (if any of them looked worse than the rest) and with 0.5 the rest of configurations. In order to measure the similarity between the system and humans, users were asked (questions 17–19) to guess, in each configuration, if their interlocutor was a human or a machine.

7.4. Texts and sessions

To perform these experiments, the texts were randomly selected from a pool of 72 texts. These correspond to simple excerpts from children's books for readers of ages 10–12. Each text had an approximate length of 200 characters (mean: 201.9; standard deviation: 20.98).

We recorded 141 sessions, each containing an average of 1 min and 42 s of dialog, totaling roughly 4 h of dialog corpus. Of this, 78 min corresponded to human turn-taking strategy, 72 with the proposed turn-taking strategy and 101 with an interaction cycle. Each subject was recorded on video. Trained annotators orthographically transcribed the recordings and aligned the words to the speech signal, yielding a total average of 50.05 words (33.66 advanced, 78.28 cycle, 38.2 human). Additionally, self-repairs, non-word vocalizations, laughs, coughs and breaths were marked as words.

8. Results and discussion

Observing the results (see Table 3), these show that the incremental processing (configuration B) significantly reduces (41.85 %) the time required to solve the dictation compared to the interaction cycle processing (configuration A). In the same way, the interaction was solved with less than half the words (43%) and the number of contributions was halved.

Regarding usability, an initial analysis of the results (see Table 4) revealed that the configuration for advanced turn-taking (configuration B) noticeably improved the interaction with respect to the interaction cycle (configuration A) for the variables satisfaction, organization, suitability, convenience, preference and co-operativity.

From an Analysis of Variance (see Table 5), results in organization of the interaction, suitability of the turn-taking strategy and convenience of the configuration to the task are significantly greater for advanced turn-taking strategy than for the interaction cycle (with p-values less than 0.005). On the same way, advanced turn-taking obtained significantly good marks for these parameters regarding the human turn-taking strategy. The mean organization perceived by the user is greater for advanced turn-taking configuration ($M = 0.46$, SD = 0.35) than for the interaction cycle ($M = 0.21$, SD = 0.30) and very close to human turn-taking ($M = 0.71$, SD = 0.34). Advanced turn-taking ($M = 0.49$, SD = 0.31) is also more suitable for solving the proposed task than the interaction cycle ($M = 0.13$, SD = 0.27) and it obtains a good mark compared with human turn-taking ($M = 0.76$, SD = 0.30). Similarly, advanced turn-taking configuration ($M = 0.54$, SD = 0.35) is perceived as more convenient than interaction cycle ($M = 0.13$, SD = 0.27) in the proposed domain, although not as much as human turn-taking ($M = 0.73$, SD = 0.30).

The advanced turn-taking strategy significantly improves the interaction cycle for the parameters users' satisfaction, users' preference and system's co-opertivity. Users' satisfaction is clearly higher for advanced ($M = 5.45$, SD = 2.29) and human ($M = 5.94$, SD = 1.95) turn-taking than for the interaction cycle ($M = 2.84$, SD = 1.92). Both advanced ($M = 0.53$, SD = 0.38) and human turn-taking ($M = 0.69$, SD = 0.34) are preferred by the users, compared

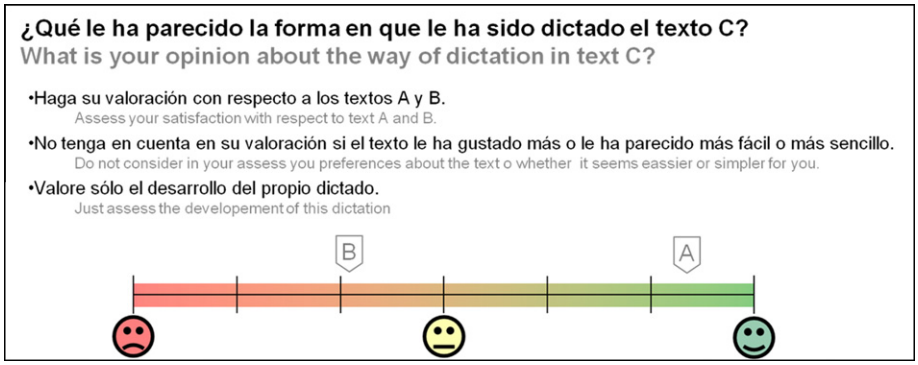

Fig. 6. User satisfaction assessment questionnaire for a turn-taking configuration.

Please answer the following questions:

	Question	Possible Answers
1	Age:	[0..99]
2	Gender:	[men; women]
3	Educational Background (Please indicate which one fits best your case):	[did not finish middle school; finished middle school; high school; superior or university studies; post-graduate studies; Information Technologies Expert; Human-Computer Interaction Expert]
4	How often do you use a computer?	[never/occasionally; usually]
5	Do any of these forms of dictation seem more mechanical than the rest? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
6	And less? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
7	Do any of these forms of dictation seem more disorganized or chaotic than the rest? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
8	And more orderly and structured? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
9	Has the leader of the experiment been more attentive and collaborative in some of the configurations than in the rest? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
10	And less? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
11	Do you believe that any of these ways of dictation makes it easier to solve the exercise than the rest? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
12	And harder? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
13	During certain dictation, have you felt more comfortable than the rest? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
14	And more uncomfortable? If so, which one? (If you doubt between several possibilities, please mark "No")	[no;A;B;C]
15	If you could choose between the different proposed dictations, which one would you pick? (If you doubt between several possibilities, please mark "None")	[none;A;B;C]
16	Would you discard any of them? (If you doubt between several possibilities, please mark "None")	[none;A;B;C]

In some cases, the dictation may not have been read by a person but a machine that prepared and displayed on screen the messages that the leader read out loud.
Before ending this exercise, please indicate if the dictation was rendered by a human or a machine:

	Question	Possible Answers
17	Dictation A	[person; machine; no response]
18	Dictation B	[person; machine; no response]
19	Dictation C	[person; machine; no response]

Fig. 7. Subjective evaluation questionnaire of different turn-taking configurations.

to the interaction cycle ($M = 0.12$, SD = 0.24). The average system's co-operativity improves with the advanced ($M = 0.51$, SD = 0.29) and human turn-taking configurations ($M = 0.56$, SD = 0.31) than with the interaction cycle ($M = 0.32$, SD = 0.31). In these cases, pairwise comparisons between advanced turn-taking and human turn-taking were non-significant. The analysis of the dynamicity did also not reveal significant results.

Table 3
Performance comparison of interaction cycle (A) and incremental processing (B).

Processing strategy	Duration (s)	Exchanged words (#)	Contributions (#)
A: Cycle	41.82	78.28	5.32
B: Advanced	**17.50**	**33.66**	**2.57**

Table 4
Means (with standard deviations, minimums and maximums) of all configurations.

	Satisfaction				Organization*		Suitability*		Convenience*		Preference*		Co-operativity*		Dinamicity*	
	M	SD	MIN	MAX	M	SD	M	SD	M	SD	M	SD	M	SD	M	SD
A: Cycle	2.84	1.92	0	7	0.21	0.30	0.13	0.27	0.13	0.27	0.12	0.24	0.32	0.31	0.44	0.33
B: Advan.	5.45	2.29	0	10	0.46	0.35	0.49	0.31	0.54	0.35	0.53	0.38	0.51	0.29	0.41	0.36
C: Human	5.94	1.95	3	10	0.71	0.34	0.76	0.30	0.73	0.30	0.69	0.34	0.56	0.31	0.53	0.36

* Min = 0 and Max = 1 in all cases.

Table 5
Analysis of Variance (ANOVA) for all configurations and pairs of configurations.

	ANOVA							
	All configurations		Pairs comparisons					
	$F_{A,B,C}$	$P_{A,B,C}$	$F_{A,C}$	$P_{A,C}$	$F_{A,B}$	$P_{A,B}$	$F_{B,C}$	$P_{B,C}$
Satisfaction	25.55	**<0.0001**	50.17	**<0.0001**	29.74	**<0.0001**	1.04	**0.311***
Organization	22.40	**<0.0001**	48.00	**<0.0001**	12.10	**0.001**	9.71	**0.003**
Suitability	43.99	**<0.0001**	92.91	**<0.0001**	28.92	**<0.0001**	14.96	**<0.0001**
Convenience	38.38	**<0.0001**	85.66	**<0.0001**	33.03	**<0.0001**	6.75	**0.011**
Preference	32.63	**<0.0001**	75.44	**<0.0001**	32.37	**<0.0001**	4.21	0.044*
Co-operativity	6.932	**0.001**	11.992	**0.001**	7.84	**0.006**	0.57	0.452*
Dinamicity*	1.17	0.315*	1.32	0.255*	0.11	0.743*	1.991	0.162*

Table 6
Left: Number of users that identify the system as machine or human for each configuration. Right: χ^2 Test for homogeneity of all configurations and pairs of configurations.

	Human similarity			χ^2 Test							
	Machine	Do not know	Human	All configurations		Pairs comparison					
				$\chi_{A,B,C}^2$	$P_{A,B,C}$	$\chi_{A,C}^2$	$P_{A,C}$	$\chi_{A,B}^2$	$P_{A,B}$	$\chi_{B,C}^2$	$P_{B,C}$
A: Cycle	21	2	16	17.51	**0.002**	10.94	**0.004**	12.76	**0.002**	0.09	0.954*
B: Advanced	6	4	29								
C: Human	7	4	28								

In short, advanced turn-taking behaves noticeably more naturally than turn-taking by interaction cycle (although in the case of dynamicity conclusive results are not obtained). In the same way, the differences found are very significant when comparing the interaction cycle with the rest of the configurations, but they are not so clear when comparing advanced turn-taking and human turn-taking.

Regarding the similarity with human turn-taking of the different turn-taking strategies (see Table 6), it can be observed that most users identified turn-taking according to the interaction cycle as a machine turn-taking (21 users, as opposed to 16 who think that it was a person and 2 who did not know). Besides, they tended to consider that the interlocutor was human in both the advanced management (29 users, compared to 6 who think that it was a machine and 4 who did not know) and the real human interlocutor (28 users, as opposed to 7 who think it was a machine and 4 who did not know). A Chi-square test of independence determined that the differences found are clearly significant in an overall analysis. From an analysis by configuration pairs, we observed that the differences found between the interaction cycle and the rest of configurations were indeed clearly significant, but the pair comparison between advanced turn-taking and human turn-taking was non-significant. Consequently, it can be stated that users are clearly capable of distinguishing turn-taking by interaction cycle from the rest of turn-taking strategies, but it is harder for them to identify the differences between advanced turn-taking and human turn-taking in restricted interaction domains (such as those evaluated here).

9. Conclusions

This paper has presented a proposal for the inclusion of the Continuity Manager and Processes Coordinator components. These components are proposed for approaching the interpretation and generation processes of natural interaction with an incremental focus. With them, it is possible to adapt the system's contribution in course to the contributions that its interlocutors develop in parallel, the simultaneous changes that occur in the circumstances that surround the interaction (session context, users, situation and emotions) and the system's proactivity. The incorporation of these components, in combination with the Turn-taking Manager and a joint-action Dialog Manager, lay the foundation for the management of phenomena of great importance in the natural development of interaction, such as overlapping turn situations (simultaneous feedback, turn-taking requests) and interruptions (caused by errors in the estimation of the state of participant turns, speech and the candidates to take the floor or fights for the turn of speech). The capacity to interpret and generate turn-taking markers that are expressed as alterations in the temporary turn continuity (through the Continuity Manager component) should also be noted.

This approach has been evaluated in a complete natural interaction system, and the results reveal improvements in both objective metrics of interaction performance and the users' subjective assessment with respect to the interaction cycle (classic approach). It reaches very close marks to human interactive abilities in aspects like dynamicity, organization, co-operativity, convenience and suitability to advanced turn-taking domains, and it shows high similarity with human interaction procedures and strategies.

The following step in our research is to study what consequences this processing approach has in the rest of the components of a natural interaction system, especially those regarding to some Interface Components like voice synthesizers and speech recognizers, evolving them as required to take advantage of the new possibilities this incremental management has in the

temporal development of the interaction. On the other hand, this paper deals with the recognition and synthesis of some common turn-taking markers (repetitions, silences, etc.), but the recognition and synthesis of the complete set is still an ongoing work.

Acknowledgements

The development of this approach and its construction as part of the Natural Interaction System LaBDA-Interactor has been partially supported by MA2VICMR (Regional Government of Madrid, S2009/TIC-1542), SemAnts (Spanish Ministry of Industry, Tourism and Trade, AVANZA I+D TSI-020110-2009-419); THUBAN (Spanish Ministry of Education and Science, TIN2008-02711); and 'Access Channel to Digital Resources and Contents' (Spanish Ministry of Education and Science, TSI-020501-2008-54).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.intcom.2012.02.003.

References

Besser, J., Alexandersson, J., 2008. A comprehensive disuency model for multi-party interaction. In: Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue.

Bunt, H., 2009. Multifunctionality and multidimensional dialogue semantics. In: Proceedings of DiaHolmia, pp.3–14.

Calle, J., 2004. Interacción Natural Mediante Procesamiento Intencional: Modelo de Hilos en diálogos. Ph.D. Dissertation (Spanish), Univ. Politécnica de Madrid, Spain.

Dybkjær, L., Bernsen, N.O., Minker, W., 2004. Evaluation and usability of multimodal spoken language dialog systems. Speech Communication 43, 33–54.

Goodwing, C., 1981. Conversational Organization: Interaction Between Speakers and Hearers. Academic Press, New York.

Jefferson, G., 1989. Preeliminary notes on a possible metric which provides for a standard maximum silence of approximately a second in conversation. Conversation, 166–196.

Kilger, A., Finkler, W., 1995. Incremental Generation for Real-Time Applications. Research Report RR-95-11, DFKI GmbH, Saarbrüken, Germany.

Komatani, K., Rudnicky, A., 2009. Predicting barge-in utterance errors by using implicitly supervised ASR accuracy and barge-in rate per user. In: Proceedings of the ACL-IJCNLP, Suntec, Singapore, pp. 89–92.

Komatani, K., Kawahara, T., Okuno, H., 2008. Predicting ASR errors by exploiting barge-in rate of individual users for spoken dialog systems. In: Proceedings of the Interspeech, Brisbane, Australia, pp. 183–186.

López-Cózar, R., Spoken, M., 2005. Multilingual and Multimodal Dialog Systems: Development and Assessment. John Wiley & Sons Publishers.

Reithinger, N., Kipp, M., 1998. Large scale dialog annotation in Verbmobil. In: Workshop Proceedings of ESSLLI 98.

Roberts, G.L., Bavelas, J.B., 1996. The communicative dictionary: a collaborative theory of meaning. In: Stewart, J. (Ed.), Beyond the Symbol Model. State University of New York Press, Albany, pp. 135–160.

Sacks, H., Schegloff, E.A., Jefferson, A., 1974. A simplest systematics for the organization of turn-taking for conversation. Language 50, 696–735.

Stivers, T., Enfield, N.J., Brown, P., Englert, C., Hayashi, M., Heinemann, T., Hoymann, G., Rossano, F., Ruiter, J.P., Yoon, K.E., Levinson, S.C., 2009. Universals and cultural variation in turn-taking in conversation. In: Kay, Paul (Ed.), Proceedings of the National Academy of Sciences of the United States of America. <http://www.pnas.org/content/106/26/10587.full>.

International Telecommunication Union (ITU-T), 2003. One-way Transmission Time. ITU-T Recommendation G. 114.

Thórisson, K.R., 2002. Natural Turn-Taking Needs No Manual: Computational Theory and Model, from Perception to Action. Multimodality in Language and Speech Systems. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 173–207.

Turunen, M., Hakulinen, J., 2000. Mailman – a multilingual speech-only E-mail client based on an adaptive speech application framework. In: Proceedings of Workshop on Multi-Lingual Speech Communication (MSC 2000), pp. 7–12.

Vanderheiden, G.C., Zimmermann, G., 2005. Use of user interface sockets to create naturally evolving intelligent environments. In: 11th International Conference on Human–Computer Interaction, Caesars Palace, Las Vegas, Nevada, USA.

Walker, M.A., Litman, D.J., Kamm, C.A., Abella, A., 1998. Evaluating spoken dialog agents with PARADISE: two case studies. Computer Speech and Language 12 (3).

Ward, C.W., Pellom, B., 2003. The CU Communicator System. IEEE Workshop on Automatic Speech Recognition and Understanding, Keystone Colorado.