IOCA: An Interaction-Oriented Cognitive Architecture

Luis A. Pineda, Ivan V. Meza, Héctor H. Avilés, Carlos Gershenson, Caleb Rascón, Montserrat Alvarado and Lisset Salinas
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS)
Universidad Nacional Autónoma de México (UNAM)
México D. F., México,
lpineda@unam.mx

Abstract

In this paper an interaction-oriented cognitive architecture for the specification and construction of situated systems and service robots is presented. The architecture is centered on an interaction model, called *dialogue model*, with its corresponding program interpreter or *Dialogue Manager*. A dialogue model represents the task structure of a specific application, and coordinates interpretations produced by the system's perceptual devices with the system's intentional actions. The architecture also supports reactive behavior, which relates context independent input information with the system's rendering devices directly. The present architecture has been used for the specification and implementation of fixed multimodal applications, and also of service robots with spoken language, vision and motor behavior, in a simple, integrated and modular fashion, where the cognitive architecture's modules and processes are generic, but each task is represented with a specific dialogue model and its associated knowledge structures.

1 An Interaction-oriented Cognitive Architecture

Autonomous systems capable of interacting with the world through language, vision and motor behavior need to be able to perform reactive and representational behaviors. Reactive behavior involves responding to world's stimuli directly in a context independent manner, while representational or "intentional" behavior involves assigning interpretations to the world's stimuli, mostly in a context dependent manner, and acting upon those interpretations. Detecting and avoiding an obstacle and turning towards a source of sound are better thought of as reactive behaviors, while reasoning, planning and problem-solving are representational processes. Reactive and representational behaviors can also be distinguished in terms of the time elapsed from the stimulus to the response: while the reactive loop is performed instantaneously, the latter can take several seconds, minutes or even longer periods of time. Consequently, several reactive behaviors can be embedded within one representational loop. Another distinctive feature between these two kinds of behaviors is that while the flow of attention, language and thought is mostly sequential, several reactive processes can be performed simultaneously, and the agent can be mostly unaware of performing these behaviors. Yet, despite all these differences, representational and reactive behavior need to be coordinated in order the agent interacts with the world in a coherent and robust fashion. The integration and coordination of these functionalities in autonomous agents requires the definition and construction of a congruent computational framework; for this, over the last few years we have been developing and testing the Interaction-oriented Cognitive Architecture (IOCA). A cognitive architecture is a system that integrates perception, thought and action, where the specific knowledge of the task and domain can vary but the computational structures and processes remain constant (e.g. [Chong *et al*., 2007]). IOCA is oriented towards the interaction between the computational agent and the world, including the interpretation of external representations (e.g. spoken language, text, diagrams, posters, etc.). The input-output representational loop involves the recognition and interpretation of the external stimuli, the selection of the appropriate action by the Dialogue Manager (DM), and its full specification and rendering.

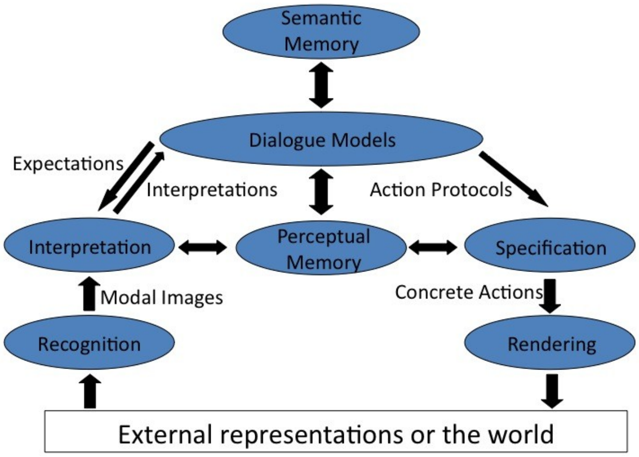

Figure 1. Interaction-oriented Cognitive Architecture (IOCA)

IOCA incorporates a semantic and a perceptual memory; this distinction corresponds loosely to the traditional distinction between semantic and episodic memory that is widely used in cognitive psychology and neuropsychology [Tulving, 1972]. The semantic memory holds concepts, particular and general, used while carrying out the task and its domain. The knowledge stored in this structure has a propositional character and is modality independent. We are using a logical representation with Prolog clauses in our current implementations, but alternative schemes, like semantic networks or description logics, could also be used.

The perceptual memory, in turn, stores associations between modality specific *internal images* or "percepts" and their corresponding interpretations or *meanings*. Internal images, on the one hand, represent the sensitive characteristics of the external stimuli and are associated to a perception modality. However, these images also capture the way the sensed information is structured (i.e. the external pattern). For instance, an object in the world, such as a diagram, a map, a text, etc., can all be perceived through the visual channel. The image, however, is "seen" differently in each case, and stored in a particular format or code, that corresponds to that particular "way of seeing". In the present framework each of these codes corresponds to a *modality*; thus, there is a modality for each "way of seeing", and each may involve one or more recognition devices (e.g. an Automatic Speech Recognition (ASR) System or a vision recognition machine). In addition, an internal image codifies the corresponding external pattern independently of its meaning. For instance, the product of an ASR system is an uninterpreted text; a SIFT vector is the product of codifying a visual image independent of its interpretation. Internal images are minimal information structures aimed to distinguish the particular concept in the input from the set of particular or general concepts in the interpretation context. Consequently, internal images do not have to be fully fleshed out representations of external objects (e.g., 2-D or 3-D geometric constructions with color or texture). The patterns represented through internal images can be dynamic and evolve in space and time. For instance, the visual pattern of a physical gesture, like "halt", that can be codified as a Hidden-Markov Model [Avilés *et al*., 2010a]. Regular expressions and specialized natural language grammars are also considered as internal images in the perceptual memory. In this view, particular or general concepts are associated to the regular expressions or grammars that "select" these concepts.

The meanings of internal images, on the other hand, are represented in a propositional format, which is modality-independent, and the expressions representing these meanings can be thought of as "tags" of the corresponding percepts. In this way, internal images can be interpreted as expressing particular or general concepts. This structure also permits to access concepts or interpretations via their percepts and vice versa. The associations between internal images and their interpretations can be established beforehand when the application is developed, or dynamically when the concept associated with the image is provided by the human user at the time the image is recognized by the system in the interactive task.

We turn now to the description of the main representational loop. The recognition modules translate external patterns sensed by the recognition devices into the corresponding internal images in the corresponding modal code, mainly in context independent way and in a bottom-up fashion.

The *interpretation module* is responsible of assigning interpretations or meanings to such internal images. This is a context dependent process that takes into account the expectations of the system that are present in the interpretation situation, as will be elaborated in Section 2. This process uses the perceptual memory and performs a qualitative match between the images recovered by the recognition devices and the images in the perceptual memory, which are stored in the same modal code. The result of the interpretation process is the "meaning" (i.e. the interpretation) associated to the external image in the interpretation context. As the number of associations in the perceptual memory can be quite large, the *expectations* of the current situations are also used as the indexes of the associations to be considered in specific interpretation situation. By this account, the expectations not only set up the interpretation context but also select the relevant memories to be used for the particular interpretation act. The recognition and interpretation levels of the architecture correspond to the overall perceptual process whose purpose is to assign interpretations to the external patterns conveying linguistic messages or events in the world that are attended to and acted upon "intentionally" by the computational agent.

The central module in the interaction loop is the dialogue model with its associated program interpreter or D*ialogue Manager* (DM). This contains the specification of the task's structure and relates expectations and interpretations with the corresponding intentional actions. Interpretations and actions at this level are specified in a propositional format that is independent of the input and output modalities.

In the output side, *actions* are performed as a response to interpretations; these can be *external*, like displaying an image, synthesizing a text or moving a robot, but also *internal*, like performing a reasoning or a planning task, involving only the representational structures of the system. In this sense, we distinguish linguistic and interaction protocols, which are stated through the dialogue models, from the "thought" processes, which are internal actions that are called upon by the dialogue model when required. Actions can be composite and involve a number of basic actions, more than one output device, and an internal and external part. Dialogue models have also access to the interaction history, and expectations and actions can be stated dynamically in terms of the events that happened before in the current task. Dialogue models can also access the knowledge stored in the semantic and perceptual memory (e.g. for heterogeneous reasoning). Finally, the action protocols specified in dialogue models are fully specified before they are sent to the specific rendering devices of the system.

IOCA differs from other cognitive architecture in that it is focused on the communication channel, and on the inclusion of a perceptual memory for the explicit recollection of sensory information. IOCA also aims to distinguish the main communication loop involving interpretations from the cognitive processes proper, and also to understand the relation between representational and reactive behavior. In doing so, IOCA focuses on the questions related to the interaction between language, perception and thought.

2 Specification and Interpretation of Dialogue Models

The central component of the cognitive architecture is the *dialogue model* –or interaction model– through which the task structure and the communication protocols between the computational agent and the human user are specified. Dialogue models are defined in relation to a basic notion: the *situation*. A situation is an “intentional state” of the agent, which is defined in relation to the expectations of the agent in the situation (either possible messages with communicative intent produced by the human interlocutor or natural event in the world), the actions the agent should perform in case a specific expectation is met, and the situations into which the agent moves after performing such action. In this way, situations are contextualized in terms of generic interaction protocols. These protocols represent the structure of the task, and traveling from the initial to the final situation corresponds to performing the task successfully.

Expectations are the set of potential speech acts types (e.g. [Levinson, 1983]) that can be expressed by the interlocutor in the situation, in addition to the potential natural events that can occur in the world in the situation, that are also handled intentionally. Expectations are expressed through statements involving the system *S* and the human user *U* like, for instance, “*S* expects that *U* commands *S* to make *p*” or “*S* expects that *U* ask *S* to provide information *q*”. However, as the expectations are embedded in the protocols and the corresponding actions assume this intentional interpretation (i.e., *S* makes *p* and *S* provides information *q*), the intentional statements are implicit in the interpretation process and only the conceptual content in the expectations is stated explicitly in the dialogue models (e.g. the propositions *p* and *q* in the examples above).

Speech acts are normally direct, in the sense that declarative statements are used for communicating facts or beliefs, interrogative for making questions, and imperatives for commands, etc., where each of these modalities of expression has a characteristic intonation. However, the basic relation between the type of intention and the modality of expression is often changed, as when a command is expressed through a polite question (e.g., "Could you show me poster A?"), producing the so-called indirect speech acts, which pose great challenges to the interpretation process. In order to interpret speech acts, either direct or indirect, we take advantage of the context present at the interpretation situation, and the interpretation problem is seen as what is the most likely intention among the expectations of the situation that is intended by the interlocutor. In this sense, expectations are conceived as *a priori* knowledge, while the input information (the actual external stimuli) is taken as evidence (i.e. likelihood) in favor of a particular expectation. The actual interpretation of the input message is the “grounded” expectation that is best met by the input information in the interpretation situation. This makes the interpretation process as a whole have a strong Bayesian flavor. However, *a priori* knowledge and likelihoods need not be numerical probabilities, as the “product operator” between these two is the interpreter, that collects the output of the recognition devices, looks up the relevant percepts in the perceptual memory, and produces the actual interpretation, expressed as a grounded speech act in the dialogue model.

Natural states and events in the world that are expected by the computational agent are also treated intentionally, and are defined as expectations of the situations in which they are likely to occur. For instance, if a robot is standing in front of a door it may have the expectation that the door is open or that it is closed. In this case, the actual image recognized visually has no communicational intent, but nevertheless it is an expectation that has to be acted upon intentionally in the context (e.g. crossing the door if it is opened or asking for the door to be opened otherwise). In this case, although the stimulus is visual, it is subject to interpretation and the behavior has a representational character.

Speech acts can express propositional or conceptual content (e.g. "Please, explain me poster A."); can assert that the message has been understood as intended (e.g. "Do you want me to explain poster A?"); and can maintain the communication channel so the interlocutors can establish and preserve a “common ground” (e.g. "I didn’t hear you, can you say it again?") [Clark and Schaefer, 1989]. The structure of practical dialogues [Allen *et al.*, 2001] oriented to solve specific tasks has been analyzed with tagging schemes that consider these three levels of speech acts (i.e. conceptual content, agreement, and communication) and the relations between a speech act and the preceding and following acts, which establishes a strong restriction in the structure of intentional transactions [Allen and Core, 1997; Pineda *et al.*, 2007]. These intuitions are also used in the specification of dialogue models: agreement and communications protocols can be stated to make sure the system and the user have a common ground. Also, whenever no expectation in a situation is satisfied, the system is out of context (i.e. the common ground has been lost) and invokes recovery protocols, stated also as dialogue models, with the purpose to set itself in context again. These protocols can also be used to restore the context when an expected natural event does not occur when it should.

The actions performed by the system in response to an interpretation are also thought of as speech acts. For the specification of these actions we follow loosely Rhetorical Structure Theory (RST) [Mann and Thompson, 1988], where an action predicate stands for a “rhetorical structure” with one or more basic actions. Each basic predicate in the structure stands for a particular action, either internal or external. For instance, an explanation may involve a presentation, an elaboration, a generalization expressed through spoken language, and even an exemplification expressed through a picture or a video. Motor actions are also stated through rhetorical structures (e.g. *move*(a, b)). Action predicates have to be fully specified, possibly using information in the perceptual memory, before the corresponding actions are rendered in an output modality.

Dialogue models have a graphical representation where situations are represented through nodes and situation relations are represented through directed links. Every link has a label of the form $\alpha\beta$, where α stands for an expectation and β stands for the action that is performed by the system when the expectation α is satisfied in the current situation s_i. As a result of performing such action, the system moves to the situation s_j at the end of the link, as illustrated in Figure 2. Situations can be basic, in the sense that a particular interpretation act takes place at the situation (e.g., through language or vision, or both). Situations are typed, and there is one type of situation for each modality defined in the perceptual memory, so the DM considers the situations type in order to select the appropriate recognition devices, with the particular modality code, to perform each basic interpretation act. There is also a special type of situation that we refer to as *recursive*, which embeds a full dialogue model. This expressive power permits to model complex applications in a simple and modular way, where composite tasks have a stack structure. The formalism corresponds to recursive transition networks (RTN), augmented with functions that permit the dynamic specification of expectations, actions and next situations. We refer to this formalism as Functional-RTN or F-RTN [Pineda, 2008; Pineda *et al*., 2010].

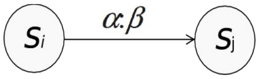

Figure 2: Graphical Representation of Dialogue Models

The conceptual content in expectations can be of three kinds, which are as follows:

(1) **Propositional:** These are concrete expectations represented with constants or saturated propositions (e.g. a, $p(a, b)$) in the dialogue models.

(2) **Predicative:** These are expectations involving a limited form of abstraction, represented as open predicates or predicative functions (e.g. $p(x)$, $q(\mathrm{a}, y)$) in the dialogue models. To meet expectations of this kind, one or more parameter needs to be extracted from the world, and these become the arguments in the expression representing the interpretation. For instance, if the robot asks the users for his or her name, the expectation is represented as *name*(x), and the interpretation of the user's reply, in case the expectation is met. For instance, "I'm Peter", is represented as *name*(*peter*). These predicates are interpreted indexically in relation to the agents involved in the transaction and in relation to the local spatial and temporal context.

(3) **Functional:** These depend of the interaction history at the level of the interpretations, actions and situations, which is collected by the system along the interaction. In the present framework, this "working memory" structure is called the *anaphoric context*. Although the task protocols are specified in advance through the dialogue models, expectations and actions can change along the task, and need to be determined dynamically in relation to the context. These kinds of expectations are represented through explicit functions in the dialogue models. These functions have the anaphoric context as one of their argument, and their values are propositional or predicative expressions representing expectations. Functional expectations are evaluated first, and their values are passed top-down to the interpreter in the current interpretation act.

The next situation in a dialogue model's transition can also depend on the anaphoric context. In this case, the situation to which the agent has to move is represented through a function h whose argument is again the anaphoric context but its value is the actual next situation. In Figure 3, the function h is represented by a small dot, and its possible values by dashed-links.

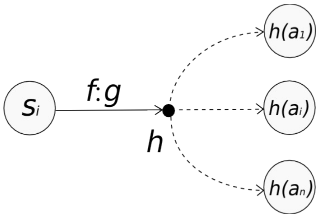

Figure 3: Functional representation of expectations, actions and transitions

Situations are also parametric objects, and their arguments can be bound with the interpretation and action predicates' arguments, allowing the establishment of co-reference relations between terms in the interaction structure.

The system's intentional actions can also be propositional, predicative and functional, and can be determined dynamically. Predicative actions can be defined through open predicates where the free variables are bound to the situation's or expectation's arguments in the corresponding transition. Functional actions can be defined through explicit functions, as it is with expectations and next situations.

Finally, the functions that define the described functional objects can access information stored in the semantic memory, which can be considered as an additional argument. Thus, functional expectations, actions, and next situations are dynamic objects that depend not only of the anaphoric context, but also on the particular and general concepts of the application task and domain.

3. Coordination of Representational and Reactive Behavior

In the architecture discussed so far, speech acts produced by the system's interlocutor and natural events in the world need to be synchronized with the expectations of the current situation in order that the computational agent can interpret them. Otherwise, the external stimuli are left unattended by the agent, even if those stimuli are defined as expectations of other situations. Most traditional applications in static worlds with a fixed interaction initiative, such as when the robot is restricted to obey user commands or the human is guided passively by the robot, can be modeled through this expectations-based architecture. However, their model is too weak for robots that need to move or navigate flexibly and robustly in a dynamic environment; in circumstances where unexpected obstacles can appear or things can be moved; or when other dynamic agents are present, such as human interlocutors taking the interaction initiative spontaneously. In order to cope with dynamic environments, IOCA needs to be extended with a set of reactive modules, which relate the input information collected by the recognition devices with the rendering devices directly. In Figure 4 it is shown where an Autonomous Reactive System (ARS) has been added. At the moment, we are considering two main ARSs: the Autonomous Navigation System (ANS) and an Autonomous Position and Orientation Source of Sound Detection System (APOS) to allow the robot to face its interlocutor reactively. This extension requires, in addition, the inclusion of a control structure for coordinating the dialogue models with the ARSs that we called *The Coordinator,* also shown in Figure 4. This figure illustrates that the main representational loop may embed a number, possible large, of reactive loops. In this respect, IOCA loosely resembles a subsumption architecture [Chong *et al*., 2007].

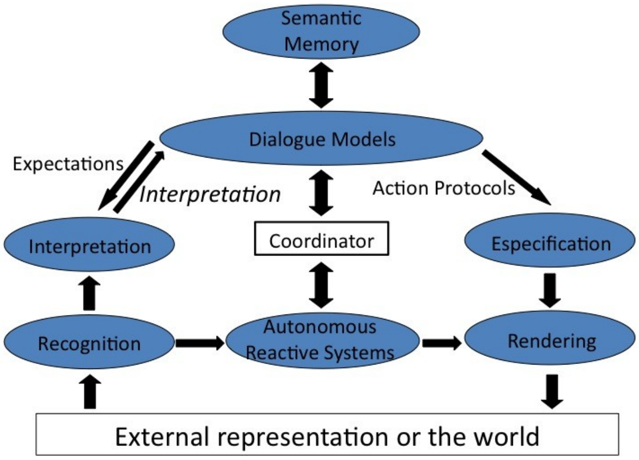

Figure 4. IOCA with Reactive Capabilities

The coordination between representational and reactive behavior is not trivial, as reactive actions can change the spatial and temporal context expected by the dialogue models, and the system needs to relocate itself in the context dynamically. In order to address this problem, we are studying three basic coordination behaviors, which are as follows:

(1) The interpretation process of the current dialogue model inside the DM and the ARSs can proceed concurrently without interfering with each other.
(2) The DM can put on hold and reactivate the ARSs, and vice versa.
(3) An ARS can load and execute a recovery dialogue model directly.

For the ARSs we are considering a basic navigation function such that given a metric map, the robot's position and orientation in this map, and a target position and orientation, the system produces and executes a plan (i.e., a sequence of moving commands) to reach the target. During this process, the system avoids obstacles reactively and adjusts its estimated position and orientation continuously in the metric map. In the present project we are focusing on the definition of the coordinator, and for the actual navigation we are exploring the use of available tools (e.g. [Vaughan *et al*., 2003]). This basic navigation functionality is called upon intentionally by an action directive stated and performed by a dialogue model; in this mode the reactive behavior is subsumed into the representational main loop in a natural way.

The APOS, in turn, monitors the acoustic environment continuously. Whenever a human voice is detected: it suspends the navigation system; turns to the interlocutor; executes a dialogue model to attend the interruption; and resumes the navigation task maintaining the original target, starting from the position and orientation that it was left after the interruption.

The coordination involves conditions in which the reactive behavior takes precedence over the representational one. For instance, imagine the robot is moving from position A to B as a result of an action request, and is carrying out a conversation with the user concurrently. In this scenario the robot has to notify the user that the navigation task has been completed when it reaches position B. To do this, the ANS has to put on hold the interpretation of the current dialogue model, make the notification, and resume the DM. Another instance in which reactive behavior takes precedence is when the APOS handles an spontaneous information request produced by the user in the middle of a moving action, which involves the interruption of both the interpretation of the current dialogue model, and perhaps of the ANS. Then, both the DM and the ANS have to be resumed when the spontaneous request has been attended, but from the context that was left after the interruption was handled.

Conversely, the coordination also involves conditions in which the representational behavior takes precedence over the reactive one. For instance, if the robot is engaged in an explanation task it may need to put on hold the APOS to avoid spontaneous distractions, and restore it when the explanation task has been accomplished. Another condition is when none of the expectations of the current situation are met, and the system has to load and execute a recovery dialogue model. For this, the system may need to suspend both the ANS and the APOS, direct all of its attention towards placing itself in context, and resume both of these when the context has been restored. Here again, the ANS has to resume the navigation task that was performing before the interruption, but from the context (i.e. position and orientation) that was left after the contingency was handled.

Finally, these generic protocols are defined in the coordinator, which controls their execution independently of the dialogue models representing the application task.

4. The robots Golem and Golem-II+

Over the last few years we have been developing the basic structure of IOCA: its dialogue model specification, interpretation theory, and programming environment. We first produced the Golem robot that was able to guide a poster session about our research projects through a spoken Spanish conversation. We also produced several applications to illustrate the integration of language, vision and navigation with Golem (e.g., [Aguilar and Pineda, 2010]). Next, we produced the application "Guess the card: Golem in Universum". It is a multimodal application in a fixed platform in a permanent stand of UNAM's science museum Universum in which the user plays a game with the system through a spoken Spanish conversation supported with computer vision and the display of images [Meza *et al.*, 2010]. Next, we presented the robot Golem-II+ which is also able to guide a poster session, but in addition to the original system, it is capable of interpreting pointing gestures expressed by the user during the interaction, illustrating the coordination between language, vision and motor behavior [Avilés *et al.*, 2010]. All of these applications have been developed using the basic representational loop only. We have also developed and tested the basic APOS algorithms with very promising results [Rascón *et al.*, 2010]. Videos of these systems are available at http://leibniz.iimas.unam.mx/~luis/. At the moment, we are incorporating and testing the extension of IOCA with reactive behaviors in the robot Golem-II+, to model the different test scenarios of the RoboCup@home competition.

Acknowledgments

We acknowledge the support of the members of the DIME and Golem group at IIMAS, UNAM. We also gratefully thank the support of grants CONACyT 81965 and PAPPIT-UNAM IN-121206, IN-104408 and IN-115710.

References

[Allen and Core, 1997] J. F. Allen and M. G. Core. Draft of DAMSL: Dialog Act Markup in Several Layers Annotation Scheme. Department of Computer Science, Rochester University, October, 1997.

[Allen *et al.*, 2001] J.F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu and A. Stent. Toward Conversational Human-Computer Interaction. AI Magazine, 22(4):27–38, Winter, 2001.

[Aguilar and Pineda, 2009] Aguilar, W., Pineda, L. A.: Integrating Graph-Based Vision Perception to Spoken Conversation in Human-Robot Interaction, J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 789–796, 2009, Springer-Verlag Berlin Heidelberg, 2009.

[Avilés, *et al.*, 2010] Avilés, H., Alvarado, M., Venegas, E., Rascón., C., Meza, I., Pineda, L.: Development of a Tour-Guide Robot Using Dialog Models and a Cognitive Architecture. IBERAMIA 2010, LNAI, Vol. 6433, Springer-Verlag, Berlin Heidelberg, pp. 512 – 521, 2010.

[Avilés *et al.*, 2010a] Avilés, H., Sucar, E., Pineda, L., Mendoza, C. A comparison of dynamic naïve Bayesian classifiers and Hidden Markov Models for gesture recognition, *Journal of Applied Research and Technology* (to appear).

[Chong *et al.*, 2007] Chong, H. Q., Tan, A. H., Ng, G. W., Integrated cognitive architectures: a survey. Artificial Intelligence Review, 28:103—130. 2007.

[Clark and Schaefer, 1989] Clark, H., Schaefer, E. F. Contributing to Discourse. *Cognitive Science*, 13:259–294, 1989.

[Levinson, 1983] Levinson, S. C. Pragmatics. Cambridge University Press, Cambridge, UK, 1983.

[Mann and Thompson, 1988] Mann, W. C. and Thompson. S. Rhetorical Structure Theory: Towards a functional theory of text organization, Text 8(3), pp. 243—281, 1988.

[Meza, *et al.*, 2010] Meza, I., Salinas, L., Venegas, E., Castellanos, H., Chavarria, A., Pineda, L.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. IBERAMIA 2010, LNAI, Vol. 6433, Springer-Verlag, Berlin Heidelberg, pp. 346 – 355, 2010.

[Pineda *et al.*, 2007] L. Pineda, V. Estrada, S. Coria y J. Allen, The obligations and common ground structure of practical dialogues, Inteligencia Artificial, *Revista Iberoamericana de Inteligencia Artificial* (2007), Vol. 11 (36), pp. 9-17.

[Pineda, *et al.*, 2008] Pineda, L. A.: Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction, in Artificial Intelligence for Humans: Service Robots and Social Modeling, G. Sidorov (Ed.), SMIA, México, pp. 33–50, 2008.

[Pineda, *et al.*, 2010] Pineda, L., Meza, I, Salinas, L.: Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI. A. Kuri-Morales and G. Simari (Eds.): IBERAMIA 2010, LNAI, Vol. 6433, Springer-Verlag, Berlin Heidelberg, pp. 20 – 29, 2010.

[Rascón, *et al.*, 2010] Rascón, C., Avilés, H., Pineda, L.: Robotic Orientation towards Speaker in Human-Robot Interaction. IBERAMIA 2010, LNAI, Vol. 6433, Springer-Verlag, Berlin Heidelberg, pp. 10 – 19, 2010.

[Tulving, 1972] Tulving, E.: Memory systems: episodic and semantic memory, In E. Tulving and W. Donaldson (Eds.), Organization of Memory. New York: Academic Press. pp. 381-403, 1972.

[Vaughan *et al.*, 2003] Richard T. Vaughan, Brian P. Gerkey, and Andrew Howard. On device abstractions for portable, reusable robot code. In Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), pages 2121-2427, Las Vegas, Nevada, October 2003.