

# Language Games for Autonomous Robots

Luc Steels, *Sony Computer Science Laboratory, Paris*

**P**et robots are currently entering the consumer market, and humanoid robots will soon follow. The ultimate success of such products will depend on our ability to resolve a key question: how we can design flexible, grounded dialogue systems for autonomous robots that permit open-ended dialogue with unprepared owners? This task is

extraordinarily difficult and poses challenges for both integration and grounding.

Here, I propose a unifying idea that meets both challenges: language games. A language game is a sequence of verbal interactions between two agents situated in a specific environment. Language games both integrate the various activities required for dialogue and ground unknown words or phrases in a specific context, which helps constrain possible meanings.

Over the past five years, I have been working with a team to develop and test language games on progressively more sophisticated systems, from relatively simple camera-based systems to humanoid robots. The results of our work show that language games are a useful way to both understand and design human-robot interaction.

## Key challenges: Integration and grounding

Human-robot dialogue requires solving many fundamental AI problems, ranging from vision and speech to action planning, plan execution, and learning. AI researchers have made important progress in most of these areas over the last 20 years, aided by major advances in sensors, actuators, and computer hardware and software. As a result, we have components today that would have been hard to imagine 15 years ago.

For example, in the mid '80s, it took Shakey<sup>1</sup> seconds to laboriously extract object segments despite a carefully constructed scene, whereas the SRI Small Vision System currently performs real-time 3D template-based object matching with stereo in an unknown real environment.<sup>2</sup> But, whatever the performance of

individual AI components, we can only obtain a complete intelligent system through integration. By overcoming individual component weaknesses using output from other components, we can create a total effect that is more than the sum of its parts.

Integration must take place at two levels. First, from a computational point of view, we must combine disparate computational processes (possibly running in parallel on distributed hardware) into a coherent global system. To do this, we need a real-time distributed operating system and a secondary layer composed of standard components specialized for robot control.

From an AI viewpoint, we must combine different tasks and methods, often using fundamentally different representations and approaches. For example, object recognition might involve techniques from instance-based learning, neural network style statistical pattern recognition, and the matching of structured symbolic representations. Dialogue requires a combination of syntactic and semantic processing, coupled to components for speech and image processing.

Language games integrate all the different activities required for effective dialogue: vision, gesturing, pattern recognition, speech analysis and synthesis, conceptualization, verbalization, interpretation, and follow-up action. The agent's scripts for playing a game not only invoke the necessary components to achieve success in the game, but also trigger learning algorithms that can help the robot fill in missing concepts or learn the meaning of new natural-language phrases.

In addition to integration, human-robot dialogue

*Integration and grounding are key AI challenges for human-robot dialogue. The author and his team are tackling these issues using language games and have experimented with them on progressively more complex platforms.*

requires grounding. Grounding relates language processing's symbolic representations to sensory-motor processing. If I tell a robot to "give me the red ball," I expect it can both detect the red ball in the environment and execute the action required to hand it to me. Although impressive natural-language interfaces and software agent technologies exist,<sup>3</sup> they do not address the grounding issue.

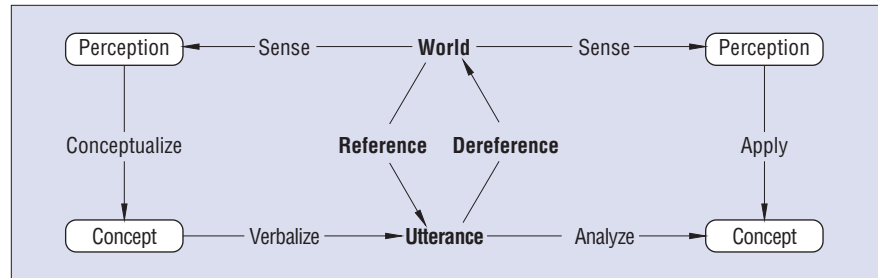
The grounding problem goes deeper than attaching labels to structures derived from signal processing and pattern-recognition algorithms. For example, color categories such as red or brown do not simply equate with wavelength measures. A fire truck, a bottle of wine, and a tomato are all "red" even though they show very different physical reflectance characteristics.

Expectations, context, and even language all influence how we categorize physical reality. Given this, we must establish a strong interdependence between the conceptual and sensory-motor layers. Pattern recognition needs guidance from the conceptual layer to avoid combinatorial explosions and the potential confusion resulting from input signals' inherent lack of detail or noise. Also, we must relate primitive concepts at the conceptual layer to sensory layer output, which often means we need techniques other than those currently employed by many purely symbolic AI programs.

Language games contribute to solving the grounding problem because they create a strong context that constrains the possible meanings of words, thus making it easier for the robot to guess unknown meanings. As an example, consider a human playing a game of showing objects to the robot, asking their name and, if the robot does not know the name, teaching it the new name. If the speaker holds up an object and says "Look, wrob," the robot can figure out that "wrob" is the name of an object, as well as being the name of the specific object that the speaker is holding. The robot can thus infer a lot from the game context and the situation that will help it learn and help it solve ambiguity and uncertainty.

### Solution: Language games

A *language game* is both an integrating glue and a vehicle for supporting the grounding of symbolic representations. The interaction between agents involves both language aspects (parsing and producing utterances) and grounding aspects. The latter aspects include sensory processing, executing gestures or actions, and, most importantly, steps for learn-



**Figure 1.** The guessing game consists of several processes. The speaker's processes are on the left, and the listener's are on the right. Between them are feedback processes that move in alternating directions until the agents settle on coherent choices for each stage.

ing new language (new words and phrases, and new meanings or pronunciations for existing words). Complex dialogue involves multiple language games interlaced with each other.

Ludwig Wittgenstein promoted the language game notion to emphasize that language and meaning are not based on context-independent abstractions, but rather arise as part of specific interactive situations. That is, the meaning of a word or phrase comes from its role in a game; for example, the meaning of "queen" in chess. This explains why a word's meaning cannot be easily defined in absolute terms but rather arises from the situation and context. It also explains why humans can easily disambiguate words or phrases. We typically interpret words in a context that strongly restricts what is being talked or written about. Our work recreates this experience for robots through situated language games.

Language games must not only be grounded and situated, but also adaptive. That is, dialogue participants must adapt their language to negotiate communications in the game. Recent research on human dialogue shows that humans invent new language while they are solving cooperative tasks.<sup>4</sup> Language is not fixed and preprogrammable, but rather highly adaptive and situated. To achieve open-ended and grounded communication with robots, we must account for this. Concept acquisition, language learning, and language negotiation must be an integral part of the dialogue.

### The guessing game

The guessing game is an example of a language game.<sup>5</sup> The "Guessing Game: A Simple Example" sidebar sketches a simplified version, and Figure 1 shows the processes it involves.

In the guessing game, the speaker tries to draw the listener's attention to an object in the environment. For example, Mary sits at the table and asks her neighbor, Pierre, for the salt by saying "salt." Mary might also point to the salt. The game's context is the table, the

objects on it, and the people around the table and their actions. The salt is the topic. From this example, it's clear that the word spoken—"salt"—is only a small part of what's going on. In addition to hearing the word, the listener must perceive and conceptualize the situation, interpret the speaker's gestures, guess what action the speaker desires, and so on. All of these elements are an intrinsic part of the language game.

The guessing game can fail in many ways. In the example above, Pierre might misunderstand the word, not know the word, or assume a different meaning for the word. This failure would become obvious in Pierre's subsequent action. He might, for example, hand Mary the water instead of the salt.

Every language game must contain provisions for detecting and repairing failure. Typically, the speaker will provide more information, possibly nonverbally through additional gestures. If failure is due to a lack of knowledge, the language game offers an opportunity for learning. For example, if Pierre does not know the word "salt" (perhaps he is French), he can use this situation to acquire a new word. If he failed to conceptualize the scene (perhaps in Pierre's native culture, salt is never purified to white grains), he can enrich his repertoire of concepts.

There are many possible variations on the guessing game, but a few factors are crucial.

1. The speaker and listener must rate different associations based on the appropriateness of their form and meaning. Doing so lets the players select the association with the highest potential for success.
2. The game must have a positive feedback loop between use and success. This lets the players use higher scoring associations in the future, increasing the likelihood of successful communication.
3. The game needs a strong structural coupling between concept formation and language, which is achieved when players have feedback on the adequacy of concepts.

## Guessing Game: A Simple Example

In a simplified version of the guessing game, utterances are single words and the lexicon consists of associations between single words and visually grounded predicates.

Each association is a triple  $\langle r, s, k \rangle$  where  $r$  is a representation,  $s$  a symbol, and  $k$  a score to reflect how successful this association has been in the past games (and hence how successful it might be in the future). Each player has his or her own lexicon. There is no global knowledge nor central control.

The steps to play the game are:

### Step 1—Shared attention

By pointing, eye gazing, moving an object, or other means, the speaker draws the listener's visual attention to the topic or at least to a narrower context that includes the topic. To aid attention, the speaker can emit a word, such as "look," and observe whether this directs the listener's gaze toward the topic. Based on this activity, we assume both agents have captured an image that reflects the shared context.

### Step 2—Speaker behavior

The speaker conceptualizes the topic, yielding a representation ( $r$ ). Conceptualization is a combination of concepts that distinguishes the topic from the other objects in the context. For this simplified game, we'll assume that the concept is a single predicate that is true for the topic but not for the other objects. For example, if every object on the table is blue but the topic is white, then color is a good way to refer to the topic. The speaker then collects all associations  $\langle r, s, k \rangle$  in his lexicon and picks out the one with highest score ( $k$ ). The symbol ( $s$ ) from this association is the best word to communicate from the speaker's viewpoint;  $s$  is transformed into a speech signal and transmitted to the listener.

### Step 3—Listener behavior

The listener receives the speech signal, recognizes the word  $s$ , and looks up all associations  $\langle r', s, m \rangle$  in her memory.

#### Step 3.1

If the listener did not have an association in memory for  $s$ ,  $s$  is a new word for the listener. She signals incomprehension, and the speaker points to the topic. In this way, the listener can concep-

tualize the scene by finding categories indicating how the topic is different from the other objects. If there is more than one possibility, she picks one, yielding a representation  $r''$  and stores the new association  $\langle r'', s, i \rangle$ , where  $i$  is an initial default score.

#### Step 3.2

If there are associations, the listener applies each representation  $r'$  to the current scene—perhaps starting with the highest scoring ones—to see whether any one picks out a unique object. If so, this is the assumed topic. There might be ambiguity due to more than one possible topic. In this case, the listener selects the referent identified by the association with the highest score. The listener then points to the topic.

### Step 4—Feedback

If the listener finds a referent (Step 3.2), there are two possible outcomes.

#### Step 4.1

The speaker agrees that the referent is correct (that it is the topic he intended) and signals his agreement. In this case, both speaker and listener increase the score of the association they used and decrease the score of competing associations. For the speaker, a competing association is one that involves the same meaning, but a different word. For the listener, a competing association is one that involves the same word, but a different meaning.

#### Step 4.2

If the speaker signals that the listener failed to recognize the topic, both speaker and listener decrease the score of the selected associations. The speaker then gives additional feedback until speaker and listener share the same topic. The listener then conceptualizes the topic from her viewpoint and either stores a new word (as in 3.2) or increases the score of an existing association (as in 4.1).

### Step 5—Acquire a new conceptualization

If the speaker or listener fail to conceptualize the scene, a concept acquisition algorithm is triggered. The attempt here is to use the situation as a learning source to acquire a new conceptualization.

## Challenges

Building dialogues for physical robots using speech input and output is quite different from building natural-language front ends for databases or expert systems, and even from building dialogues for synthetic characters.<sup>3</sup> On the positive side, physical robots have real-world situations that can help constrain the meaning of utterances, and human users can help delineate dialogue using prosody and gestures.

On the negative side, there is enormous uncertainty at every step of the process:

- Getting speakers and listeners to share attention on the same object or event is extraordinarily difficult, but is required if they are to discuss such objects or events in the real world.
- Given that they have different perspectives, speakers and listeners typically derive different low-level features and even different segments.
- Speakers and listeners cannot telepathically discern each other's conceptualizations of the world. Because there are

many ways to conceive reality, it is almost certain that conceptualizations will differ. For example, "the wine glass on the edge of the table" might also be conceptualized as "the glass from which you just drank" or simply "your glass."

- There are many ways to express conceptualizations, and language is ambiguous—a single word can have different meanings. This creates uncertainty for listeners. Moreover, we cannot assume that speakers and listeners have exactly the

same knowledge of the language. People typically have different histories of language exposure, and thus use language in subtly different ways.

- Finally, there is an inherent uncertainty and ambiguity in the speech signal itself. As decades of speech recognition research have shown, utterances articulated by the speaker are seldom unambiguously recognized by the listener.

It follows that we cannot view the different process steps outlined in Figure 1 as sequential. They must work as a dynamic constraint-propagation process in which information flows in all directions until speaker and listener settle on a coherent communication. To select the best conceptualization and verbalization, the speaker must take the listener's circumstances, prior knowledge, and viewpoint into account. The listener must perform bottom-up and top-down processing to maximize the use of all available constraints.

## Applications

For the past five years, I have been working with a team to concretize and elaborate the language game idea. Our experimental platforms have been different generations of Sony robots: the EVI pan-tilt camera and the AIBO dog-like robot. Our current work focuses on the humanoid Sony Dream Robot (SDR). All the robots use Aperios<sup>6</sup> as their real-time OS and Open-R<sup>7</sup> as the secondary computational layer.

To integrate the AI components, we designed a cognitive architecture, Coala, to run on top of these components. In keeping with AI tradition, Coala captures an agent's interaction with the world and other agents using schemas. A schema contains local slots, a schema applicability monitor, constraints, and actions specified as augmented finite state machines. Coala has facilities for memory storage and retrieval, action selection, and flexible schema execution. To interface between cognition and sensorimotor intelligence, Coala uses shared data structures. Although computational and cognitive architectural layers are critical for a successful integration, here I focus on our general design philosophy for using these tools to achieve coherent, global interactive behavior from disparate components.

## Talking Heads

The Talking Heads experiment<sup>8</sup> was one of our first. It involved two robots playing a



**Figure 2.** Setup for the Talking Heads experiment. Two pan-tilt cameras look at a white board containing colored geometric figures, which the robots use as subjects of a guessing game.

guessing game. As Figure 2 shows, each robot consisted of a pan-tilt unit with a color camera, oriented toward geometric figures pasted on a whiteboard. The situation's context consisted of a small area on the whiteboard. The topic was one of the figures, such as a red square.

For conceptualization, we used a decision-tree-like algorithm. Input to the decision-tree is output from a battery of statistical pattern-recognition and computer-vision algorithms. Thus, "left" meant that the  $x$  coordinate of the figure's middle point was less than the average  $x$  coordinates of all figures in the context, "right" meant that it was greater than this average, "large" meant that the figure's size was greater than the average size of all figures, and so on.

For concept acquisition, we used a selectionist learning method: decision-trees grow in a random fashion when the agent fails to find a concept that distinguishes the topic from other objects, and branches are pruned if they prove irrelevant or unsuccessful in subsequent language games.<sup>5</sup> The "Talking Heads Example" sidebar shows part of a game, where the listener acquires a new word for a particular shade of blue.

We set up a teleportation facility that let us run several guessing games through Internet-connected installations in different real-world locations. The whiteboard in each location featured a different configuration of colored figures. An agent population traveled through the Internet and installed themselves into different robot bodies to play the games.

People created the agents through the World Wide Web ([www.csl.sony.fr/talking-heads](http://www.csl.sony.fr/talking-heads)). Owners could also play guessing games with their agents through recorded images, and thus teach them new words. If no new word was available, a speaking agent could construct a new word. As a result, a new language progressively developed that was only partly influenced by human language.

During a three-month period, the agents played close to half a million language games and created a stable core vocabulary of 300 words (they generated thousands of words overall). Our experiment showed not only that the language game approach is useful for implementing grounded dialogues between one human and a robot, but also that the game might be useful as an explanatory model for how language originates. Indeed, an evolving population of grounded agents developed, from scratch, a repertoire of concepts and a lexicon to communicate about their environment.<sup>8</sup>

## Talking to AIBO

More recently, we used the language game framework to run experiments on the AIBO robot. In many respects, these experiments were a giant step beyond the Talking Heads experiment. AIBO is a fully autonomous mobile robot with more than a thousand behaviors, coordinated through a complex behavior-based motivational system.<sup>7</sup> Given this, getting the robot's attention and achieving shared perspective on the world prior to a game is complex. We enhanced the dialogue using the robot's gestures and movements and

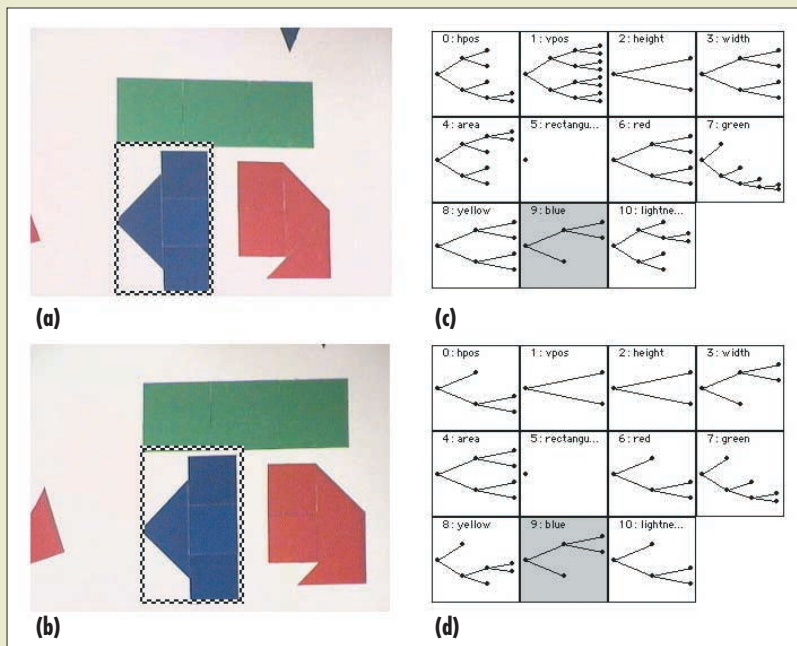
## Talking Heads Example

Figure A shows an example of the Talking Heads experiment. Table A shows the raw data that the speaker derives from the image:  $X$  is the horizontal position of figure's middle point,  $Y$  is the vertical position,  $H$  the height,  $W$  the width,  $A$  the angularity,  $R, G, Y, B$  the color opponent channels (red, green, yellow, blue), and  $L$  the brightness.

The first object (with coordinates 0,1) is the topic. Based on the decision trees, the agent conceptualizes this object in terms of the blue channel. A shade of blue (between 0.25–0.5) is distinctive for the topic, but not for any other object in the context. The speaker has three words in the lexicon for this: *Xagadude* (score 0.1), *Nibidesu* (score 0.0), and *Tetipi* (score 0.0). The speaker chooses the first word.

The listener does not know this word and performs a categorization, which happens to yield the same conceptualization. The listener therefore adds a new association to the lexicon.

Note that the listener might have easily chosen another conceptualization for this scene, such as one based on brightness or height. This would create a divergence in the lexicon. This divergence would show up in a later game when the same agents are confronted with a similarly unclear situation.



**Figure A. Example of a guessing game played by two “talking heads.” (a) the images captured by the speaker and (b) listener. Notice that they are not exactly the same, nor are (c) the speaker’s and (d) listener’s decision trees. Although their repertoires are not the same, both agents in this case chose the same distinction.**

**Table A. The raw data that the speaker derives from the image in Figure A.**

Object	$X$	$Y$	$H$	$W$	$A$	$R$	$G$	$Y$	$B$	$L$
0 (0,1)	0.37	0.71	0.48	0.21	0.45	0.17	0.00	0.00	0.39	0.28
1 (1,0.96)	0.70	0.69	0.38	0.22	0.45	0.98	0.00	0.52	0.00	0.36
2 (0.42,0.0)	0.51	0.31	0.21	0.51	0.70	0.00	0.99	0.73	0.00	0.46

its onboard visual processing and sensing. We used off-the-shelf speech components for speech input and output. Obviously, using spoken language increases still further the communication uncertainty.

Nevertheless, we successfully implemented several language games, starting with the guessing game. Figure 3 shows an example of an interaction; the “AIBO Game Dialogue” sidebar shows the dialogue. Rather than using decision trees as before, we performed conceptualization using a nearest-neighbor algorithm with a memory of stored object views. We used instance-based learning to build the object memory. Every language game was thus an opportunity to acquire a new object view or to learn about a new object class, of which the current exam-

ple was a first instance. Therefore, in this experiment we showed that any kind of concept acquisition can be used. The topic might be a single object, or it might also be an action or property of the situation. Other games focused on naming body parts and actions to be used in commands.

### Current work: Communicating with humanoids

Designing a single language game is a very difficult task. The challenge lies in many subtle details. First, you must set up the right opportunities for the robot to get every possible piece of information from the environment. You must then exploit this information to improve the robot’s understanding, and help it learn new concepts and language

whenever possible. Designing systems that can handle multiple language games is even more difficult, because humans smoothly switch from one game to another, often without a clear explicit indication.

Our current work focuses on dialogues for multiple language games within the context of humanoid robots, specifically the Sony SDR. The SDR humanoid robot has the necessary behavioral capacities to support fully animated grounded communication (gestures, visual recognition, speech input and output, and so on). The main challenge is to bring all these subsystems together into a coherent dialogue.

Managing flexible dialogues in real-world interactions is a problem similar to the action-selection problem, which has been a



**Figure 3. A guessing game between AIBO and a human experimenter, Frédéric Kaplan, involving the use and acquisition of a word for “ball.” The “AIBO Game Dialog” sidebar gives an example of dialogue from the interaction.**

central challenge in behavior-based robotics architectures.<sup>9</sup> As with action selection, robots in dialogue must swiftly respond to impulses coming from the environment, while maintaining behavioral coherence and remaining on target for long-term goals. In her work with Kismet, Cynthia Breazeal<sup>10</sup> offered a first example of how we can use ideas from behavior-based robotics to regulate human–robot interaction. The key is to introduce continuously varying motivational states and combine them in a dynamical system with sensory-motor states and behavioral progress monitors. Motivational states indicate a behavior’s desired intensity and degree of opportunity. Action selection occurs through a kind of bidding process in which different behaviors compete for execution. Motivational dynamics are tightly coupled to the environment so that the robot can swiftly switch to another behavior as required by the environment.

We are using these same principles to manage multiple language games. A game’s implementation schema has associated monitors that determine whether the schema would be appropriate in a given situation. Opportunity is determined both by environmental conditions and fragments of language utterances that signal certain games. We implement each language game using a schema or a small hierarchy of schemas, and local states monitor opportunity or progress. Games have associated motivational states that permit flexible decision and flexible

switching from one game to another.

The increased complexity of individual language games and the need to coordinate multiple language games raises the issue of syntax. Language games use and produce bits of syntactic structure as needed. Thus, there is no separate central language module that provides complete parses or handles the planning of complete sentences. Rather, syntactic processing is similar to object tracking: It is an ongoing process that yields occasional results that are immediately used to further the game. Alternatively, the game can provide strong constraints to aid syntactic processing.

It’s now becoming possible for humans to have open-ended dialogues with physically embodied robots. However, such dialogues remain extraordinarily difficult to implement, mainly because they require the integration of a wide range of technologies and methods into a coherent system.

As my examples here show, the language game metaphor is a useful way to conceive and design open-ended dialogues. Language games group all aspects of verbal interaction: the parsing and production of utterances, as well as the grounding in sensory-motor intelligence, nonverbal gestures, and actions that result from communication. In this way, language games provide the glue to integrate many diverse components into a single

## AIBO Game Dialog

The AIBO robot is preprogrammed to respond to many action names. At the start of the AIBO experiment, the experimenter tells AIBO to sit, making concentration on the language game easier.

1. *Human: Sit.*

2. *Human: Sit down.*

The experimenter then shows AIBO the ball (see Figure 3).

3. *Human: Look.*

4. *Human: Ball.*

The word “look” helps AIBO focus on the guessing game based on visual input. The robot performs image capturing and segmentation. The game is possible if AIBO finds a segment. It tries to recognize the object using a nearest-neighbor algorithm.

5. *Aibo: Ball?*

The robot asks for feedback of the word to make sure it understood it.

6. *Human: Yes.*

With this positive feedback on the pronounced word, the game proceeds. “Ball” is the word that AIBO will then associate with the object; if it is a new word, the robot will store it. If the word already exists, the robot will increase the word’s score.

whole. Language games also further language and concept acquisition, which can help robots fill in gaps and negotiate communication conventions.

The difficulty in setting up adequate learning is not in finding good machine-learning algorithms (plenty exist now) but in setting up the right opportunities for agents to learn. Properly designed language games create this opportunity. ■

## Acknowledgments

The technical work described in this article involved many people from the Sony Computer Science Laboratory in Paris, the Digital Creatures Lab in Tokyo, and the VUB Artificial Intelligence Laboratory in Brussels. In particular, Frédéric Kaplan of Sony CSL was chief architect of the language games on the AIBO. In addition, I thank Tony Belpaeme, Edwin de Jong, Toshi Doi, Masahiro Fujita, Angus McIntyre, Joris Van Looven, and Paul Vogt.

**References**

1. J.J. Nilson, "Shakey the Robot," tech. note 323, SRI AI Center, Menlo Park, Calif., 1984.
2. D. Beymer and K. Konolige, "Real-Time Tracking of Multiple People Using Continuous Detection," *IEEE Frame Rate Workshop, 1999*; available at [www.eecs.lehigh.edu/~tboult/FRAME/Beymer](http://www.eecs.lehigh.edu/~tboult/FRAME/Beymer) (current 31 Aug. 2001).
3. N. Badler, M. Palmer, and R. Bindiganavale, "Animation Control for Real-Time Virtual Humans," *Comm. ACM*, vol. 42, no. 8, Aug. 1999, pp. 64–73.
4. H.H. Clark and S.A. Brennan, "Grounding in Communication," L.B. Resnick, J.M. Levine, and S.D. Teasley, eds., *Perspectives on Socially Shared Cognition*, APA Books, Washington D.C., 1991, pp. 127–149.
5. L. Steels, "The Origins of Syntax in Visually Grounded Robotic Agents," *Artificial Intelligence*, vol. 103, nos. 1–2, 1998, pp. 133–156.
6. Y. Yokote, "The Apertos Reflective Operating System: The Concept and Its Implementation," *Sigplan Notices*, vol. 33, no. 10, 1992, pp. 414–434.
7. M. Fujita and H. Kitano, "Development of an Autonomous Quadruped Robot for Robot Entertainment," *Autonomous Robotics*, vol. 5, 1998, pp. 1–14.
8. L. Steels et al., "Crucial Factors in the Origins of Word-Meaning," *The Transition to Language*, A. Wray et al., eds., Oxford Press, UK, to be published 2002.
9. L. Steels and R.A. Brooks, eds., *The Artificial Life Route to Artificial Intelligence: Building Embodied Situated Agents*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1995.
10. C. Breazeal, "A Motivational System for Regulating Human-Robot Interaction," *Proc. AAAI98*, AAAI Press, Menlo Park, Calif., 1998, pp. 31–36.

For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dilib>.

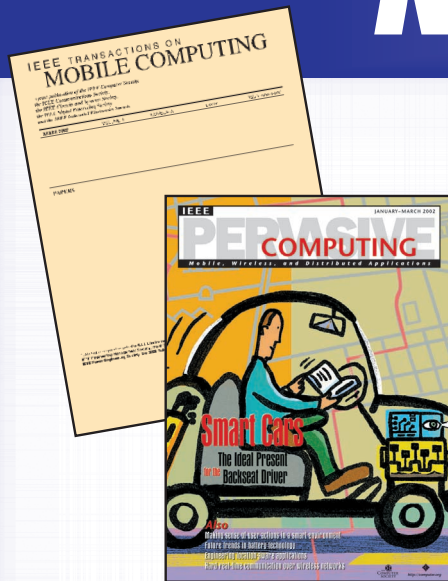
**The Author**



**Luc Steels** is professor of artificial intelligence at the University of Brussels and director of the Sony Computer Science Laboratory in Paris. His research interests in AI include robotics, vision, learning, and natural language, with a focus on the development of computational and robotic models for studying the origins of language. Contact him at AI-lab (ARTI), Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium; [steels@arti.vub.ac.be](mailto:steels@arti.vub.ac.be).

# New for 2002!

Announcing 2 New Resources for Mobile, Wireless, and Distributed Applications



The exploding popularity of mobile Internet access, third-generation wireless communication, handheld devices, and Bluetooth have made pervasive computing a reality. New mobile computing architectures, algorithms, environments, support services, hardware, and applications are coming online faster than ever. To help you keep pace, the IEEE Computer Society and Communications Society are proud to announce two new publications:

**IEEE Pervasive Computing**

Strategies for mobile, wireless, and distributed applications, including

- ▶ Mobile computing
- ▶ Wireless networks
- ▶ Security, scalability, and reliability
- ▶ Intelligent vehicles and environments
- ▶ Pervasive computing applications

<http://computer.org/pervasive>

**IEEE Transactions on Mobile Computing**

State-of-the-art research papers on topics such as

- ▶ Mobile computing
- ▶ Wireless networks
- ▶ Reliability and quality assurance
- ▶ Distributed systems architecture
- ▶ High-level protocols

<http://computer.org/tmc>