ELSEVIER

Robotics and Autonomous Systems 43 (2003) 163–173

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Shared grounding of event descriptions by autonomous robots

Luc Steels [a,b,*], Jean-Christophe Baillie [a]

[a] *SONY Computer Science Laboratory, Paris, France*
[b] *Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Brussels, Belgium*

Abstract

The paper describes a system for open-ended communication by autonomous robots about event descriptions anchored in reality through the robot's sensori-motor apparatus. The events are dynamic and agents must continually track changing situations at multiple levels of detail through their vision system. We are specifically concerned with the question how grounding can become shared through the use of external (symbolic) representations, such as natural language expressions.

Keywords: Autonomous robots; Event descriptions; Open-ended

1. Introduction

The work reported in this paper is part of a larger research effort towards grounded open-ended self-generated communication among robots [22] and grounded open-ended natural language-like communication between humans and robots [19]. *Grounded* means here that the communication is about the shared environment in which speaker and hearer are situated and which has to be perceived and interpreted autonomously by both participants through a perceptual apparatus. This contrasts with natural language interfaces to purely symbolic information systems like databases [3] or communication systems for software agents [4] in which the agents have full access to an accurate and complete representation of the environment and each others' internal states. *Open-ended* means that the communication conventions are not fixed in advance but they are negotiated and adapted to suit the communication needs of the partners. This appears necessary because human communication is open-ended as well. Humans may invent new meanings and new expressions for these meanings or adapt existing expressions to serve new purposes as part of a normal conversation [9].

Grounded verbal communication is an enormously challenging task which requires the integration of many capabilities, including speech and language processing. We believe that there is no single sweeping principle that will make a non-grounded AI system grounded. It is definitely not the case that one can simply attach a vision/action module to a logic-based reasoning system to obtain a grounded agent, nor that one can simply put a conceptual system on top of a behaviour-based robot. Instead, grounding is a matter of embodiment and very careful design, as well as tight integration of many components at many different levels. Nevertheless, it is possible to identify a set of issues that need to be dealt with and general design principles or strategies.

This paper starts with an introduction into the grounding issue in an attempt to clear up possible terminological confusions. It then describes very briefly a system that we have built which is a combination

* Corresponding author. Present address: SONY Computer Science Laboratory, Paris, France.
E-mail address: steels@arti.vub.ac.be (L. Steels).

0921-8890/03/$ – see front matter
PII: S0921-8890(02)00357-3

and integration of two subsystems reported on earlier: the PERACT system [5], designed for the visual recognition of actions, and the EVOLAN system [20], designed for exploring the symbolisation of event description in multi-agent simulations. This paper focuses mainly on the visual processing and event categorisation that anchor symbols in the world. We next turn to a discussion of the underlying design principles and end with some conclusions.

2. Terminological issues

There has been a big debate in AI and cognitive science on whether intelligence requires symbolic representations [8], and if so how these representations are supposed to be related to the world through a sensory apparatus and how this relation is acquired [11,13].

For the purposes of clarifying this discussion, we have found it useful to make a number of new distinctions. Generally speaking, representations code meanings, i.e. features of the environment relevant to the agent. We can distinguish internal and external representations. Internal representations occur when the physical structures are located inside computer memories or in brains. External representations are physical structures outside the individual: marks on a piece of paper, sounds, gestures, objects. Communication between two agents always requires external representations.

Another useful distinction which has been introduced by Sperber and Wilson, is that between Shannon coding and inferential coding, which gives rise to a distinction between information representations (I-representations) and expressive representations (E-representations), respectively. Natural language utterances are clear examples of inferential coding and thus of E-representations [18]. The interpreter is assumed to be intelligent and capable to infer meaning from mere hints. As a consequence the representation can be more compact because the interpreter shares sufficient context and background knowledge to make the appropriate inference. Most importantly, the representation need not be exclusively based on established conventions but can be the outcome of a negotiation process. Thus analogy is heavily used in natural language to invent a way for expressing a new meaning. For example, soon after Douglas, Engelbart developed the new concept of an "*x*–*y* position indicator for a display system" in the form of a box rolling over the table, it was called a mouse by analogy with the shape of a real mouse and now the whole world calls it that way. This shows that inferential coding can potentially express an open-ended set of meanings because the coding conventions can be adapted as the needs arise.

The information structures typically used in computers are examples of Shannon codings, and further called I-representations. No intelligent interpreter is assumed and so interpretation is straightforward and automatic. There is even a question whether one can speak of interpretation. All the information is in the message itself and the coding is fixed. There is no need to go through a complex process of disambiguation or the guessing of meaning. The production and interpretation of E-representations clearly requires information processing, both to code the meaning that needs to be expressed and the partial structures (such as syntactic structures) generated as part of the production and interpretation process (which sometimes even involves a model of the listener). But this does not necessarily imply that the brain internally uses E-representations. We do not want to go into that discussion here, except to point out that often philosophers, anthropologists, artists, etc. use the term representation in the sense of (external) E-representation whereas computer scientists or AI researchers use it in the sense of (internal) I-representations. Our more precise terminology is proposed to avoid this confusion.

The term grounding applies to all possible representations, and the opposite of a grounded representation is a formal representation, like an uninterpreted algebra. A grounded representation has intentionality; it is about objects and situations in the world. This implies that the agent needs processes to establish and maintain this relation. For example, there might be internal I-representations in the form of data structures (or states in neural networks) that code for the colour, position, shape, size, trajectory, speed of movement, etc. of an object in the world and these could be constructed and maintained by a vision system that is segmenting images, tracking them, and computing their properties in real time. External representations could also be grounded, in the sense that a description produced by one agent could be about an object or

situation in reality and the other agent has to ground the meaning of this description in his own perception of reality in order to understand it.

A key issue, and the one we try to solve in the experiments discussed in this paper, is how *shared grounding* can occur. This is a big problem because the agents do not have access to each other's internal states (i.e. each other's internal representations). We argue that shared grounding can be established through a negotiation process embedded in language games. So the 'symbols' that we are trying to see grounded are external symbols used in language-like communication, they are not internal symbols used in some cognitive process. Not only are we interested in single symbols (words) but also, and particularly, in the shared grounding of the meaning of grammatical structures. By negotiation we mean that agents invent representational conventions, try them out with others, and adapt their set of conventions based on the feedback on success or failure in communication.

There has been quite a lot of work (as illustrated by this special issue, as well as the papers in [10]) on how a single agent can ground his internal I-representations in reality by a sensory-motor apparatus. But there has been little work so far on shared grounding through external E-representations, i.e. how a population of agents, each with a grounded representation system, can evolve agreement on how their respective internal representations are coordinated through external representations. Other papers describing our approach (see, for example [19,22]) have focused mainly on the language part, whereas this paper focuses exclusively on the anchoring components, i.e. the vision and tracking system that generates the internal representations to be expressed.

3. Grounded language communication

The robotic installation used for the present paper is displayed in Fig. 1 and similar to that used in earlier 'talking heads' experiments [22]. It consists of two SONY pan-tilt cameras (EVI-D31) each connected to a computer, which runs the PERACT system. The computers are Bi-Xeon 1.7 GHz machines running Linux Redhat 7.1. The language-specific aspects of the system (parsers, producers, etc.) run on a third computer (Mac G4 with Common LISP) with communication through a local area network.

Fig. 1. Robotic installation used for the experiments reported in this paper. It consists of two steerable cameras capturing images of dynamic scenes. The captured images are shown on separate monitors.

3.1. Language games

The robots engage in interactions which we call language games [19]. A language game is a routinised, situated interaction between two agents. The interaction not only involves verbal communication, i.e. the parsing and producing of utterances, but also the grounding of internal representations through sensory processing, and, most importantly, steps for learning new aspects of language if necessary: new words, new meanings for existing words, new phrases. We believe that human–robot communication is best structured in terms of language games because human language interpretation requires a strong sense of context. The utterance does not contain all the information necessary for its interpretation. Words are ambiguous, many things are left unsaid, and the speech signal is notoriously difficult to decode. Because the language game makes the communication more predictable and provides a framework for semantic inference, it produces the strong constraints needed to make verbal communication doable and enables social learning [21].

The present paper focuses on one game only, namely a description game in which one agent (the speaker) describes to another agent (the hearer) an event in the world. The hearer gives feedback whether he agrees with the description or not. Some snapshots of a typical example of a scene is shown in Fig. 2. There is a hand which moves towards a (red) object and picks it up. An adequate description is: “The hand picks up the red object”. Notice that the background consists of an unaltered typical office environment with different sources of light (daylight and artificial light). The action takes place at a normal pace and the dialog takes place as a commentary on the actions in real time.

Additional typical events handled by the system are: the red ball rolls against the green block. The hand slides the pyramid against the blue box. The hand puts the red cube on top of the green one. A yellow ball rolls down a ramp. Because the world consists of dynamically changing situations, classified as events, this work is strongly related to other research on visual event classification [14,17], temporal world modelling [2], and the conceptual analysis of event expression in natural languages [23].

3.2. The semiotic cycle

To play a description game requires that the speaker perceives the situation by capturing streams of images with the camera, represent the result of sensory processing as a series of facts in memory, and then conceptualise the event and the objects in terms of roles and event types. Next the speaker must map this conceptualisation into an utterance, which includes choosing words for the predicates identifying the objects and the event, and applying the rules of grammar. The hearer must lookup the words and decode the grammatical structures, reconstruct a semantic structure, and interpret it in terms of his own world model. The language game succeeds if the utterance produced by the speaker describes an event in the recent past. The whole process is called the semiotic cycle (an extension and adaptation of the well-known ‘semiotic triangle’) and displayed in Fig. 3.

The internal conceptual representation consists of a series of facts represented in first-order predicate–calculus, following standard practices in symbolic AI [15]. A typical set of facts generated from visual processing for an event in which a red ball moves away from a green ball is:

Fig. 2. Snapshots of a typical event handled in the experiment: a hand grasping an object.

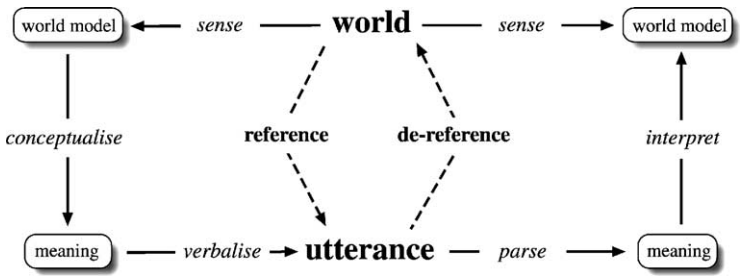

Fig. 3. The semiotic cycle: left, processes carried out by the speaker; right, processes carried out by the hearer.

```
(move-away ev-1) (move-away-patient
   ev-1 obj-1)
(move-away-source ev-1 obj-2)
  (green obj-2)
(ball obj-2) (hand obj-4) (red obj-1)
  (ball obj-1)
(larger obj-1 obj-2) (box obj-3)
  (next-to obj-1 obj-3)
```

Speaker and hearer see the situation from different angles, which often implies that there is no complete equivalence in their world models.

Each fact has three additional information items:

- A *time stamp* indicating when the fact arrived into memory. This is used for implementing forgetting: after a certain time period 'old' facts are erased and can no longer be the subject of a language communication.
- A *time period* which specifies the start and end point of a fact (when known). This makes it possible to use the temporal interval calculus [2] for representing and reasoning about actions and time.
- A *certainty* indication assigned by the vision system to this fact.

Before making an utterance, the speaker chooses (randomly) a recent event plus a set of objects related to this event (for example, ev-1, obj-1 and obj-2). For each object and for the event itself, the speaker then seeks a predicate or conjunctive combination of predicates, that are distinctive for the object or the event. Distinctive means that the predicate (or combination) is only valid for the intended object but not for any other object in the context. Thus (ball obj-3) is not distinctive for obj-3 in the above example, because both obj-1 and obj-2 are described as such. However, (green obj-3) is distinctive because obj-3 is the only green object in the context. The lexicon associates words with predicates (for example, the word "green" with the predicate 'green') and grammatical rules map additional aspects of meaning such as the predicate–argument relations into syntactic structures. The speaker assembles all of this into a complete utterance and the hearer uses the same rules in reverse to come up with a semantic structure.

The semantic structure as reconstructed by the hearer from parsing the utterance consists of a predicate–calculus expression with variables that can be matched against the facts in fact memory, again following standard practises in natural language semantics. For example, for the utterance "the red ball moves away", this expression looks as follows:

```
(move-away ?event) (move-away-
   patient ?event ?object)
(red ?object) (ball ?object)
```

When this expression is matched against the fact memory shown earlier, a unique coherent set of bindings of all variables is obtained:

```
((?event . ev-1) (?object . obj-1))
```

This is considered as an appropriate interpretation and therefore the game succeeds. The game fails when there is no such interpretation or when there is more than one set of possible bindings for all the variables.

This paper does not further discuss the (very complex) language component, nor how words or grammatical rules are invented and learned as part of the game (see [19] for more details). Instead, we focus on how agents establish the relation between the real

world as captured by the cameras and their internal world models.

3.3. Issues in grounding

There are a number of very difficult grounding issues which need to be handled within the context of this application:

- First of all, the environment which generates input to the system consists of the dynamically changing unpredictable real world. Agents have to keep up with the dynamics of the environment and produce responses within available sensory and computational resources. It follows that not just anything can be computed but resources need to be allocated in a dynamic fashion depending on the requirements of the communicative situation.
- Second, because the images are unconstrained, with natural and changing daylight, the results of visual processing are necessarily going to be noisy. For example, segments found on the basis of colour segmentation may suddenly disappear or change when light conditions are slightly changing, a condition in the world (like a hand touching an object) may during a short instant of time change because of unstable segmentation, part of an event may not be perceived due to failures in lower level visual processing, etc. So we need a way to handle noise, for example by using top-down expectations.
- Third, because the communication is open-ended, it will be necessary to adapt the visual processing to the needs of the communication partners, which implies that at least some of it will have to be learned. Another thing which has to be learned is what the vision system 'tells' the language system.

4. Visual processing

We now focus on the set of visual interpretation processes that the agents use to relate the external dynamically changing real world with the internal conceptual world models and with the meaning of natural language expressions. Rather than detailing the many vision algorithms that have been used (which are most of the time well-known state of the art algorithms [16]), we focus on the general architectural principles. More information on the PERACT system can be found in [6].

The vision system can be decomposed into three subsystems. The first one attempts to detect and track visual units at different hierarchical levels. The second subsystem detects and tracks events, again at different hierarchical levels. The third subsystem consists of feature detectors that attempt to find qualitative descriptions for units at different levels of the object or event hierarchy. The result of all these processes is a set of streams, reporting objects and their properties dynamically in response to a changing world. There is a (short term) memory of these streams that is kept as they unfold. This is called the visual memory. Some of the descriptions flow automatically into the robot's fact memory (particularly those that are at a higher level and whose certainty is beyond a threshold) and these are used by the conceptualisation system to construct or interpret semantic structures.

4.1. Detecting and tracking spatio-temporal units

The first step in grounding is to detect and 'latch onto' regions in the image that are generated by objects of interest in the environment. This results in a deictic representation [1] which establishes and monitors indexical references between internal symbols and external objects. A first innovation of the work presented here is that this tracking not only takes place for a single object, but for an open-ended set of objects at different hierarchical levels—as long as they are part of the same spatio-temporal context. The detection and tracking of units at different hierarchical levels constitutes the backbone of the vision system. It starts in a bottom-up manner from the images captured by the camera, and goes through various processing steps, first to extract spatial regions, and then to connect them in time to get spatio-temporal continuities (see Fig. 4).

More concretely the following layers are present:

(1) *Image streams*: At the first bottom layer, there is an influx of images (at a rate of 24 s and with a size of 160 × 120 pixels) supplied by the camera in the LUV colour space.
(2) *Figure/ground separation*: The next step is to identify regions that may correspond to objects in the scene, thus distinguishing figure(s) from

Fig. 4. Flow of treatment from raw image to segmented objects.

background. This is currently implemented by comparing the captured image with a stored image of the background. All zones where the pixels are not identical to the background are marked as zones of interest. This avoids processing the complete image in subsequent layers. The background needs to be learned prior to further visual processing but is updated whenever there are significant changes.

(3) *Occupancy grid*: Constructing an occupancy grid is a well known technique used in mobile robotics for navigation and path planning. An occupancy grid is a cellular representation of the environment that contains in each cell information about the probability that there is an object present [12]. We use a similar technique here. Prior to routine visual processing, probabilistic colour histograms are learned for each of the possible objects that may appear in a scene [7]. These histograms make it possible to calculate the probability that a certain pixel belongs to a particular object. The occupancy grid collects these probabilities for all pixels in each zone of interest and assigns the pixel to the object with the highest probability, if it is greater than a minimal threshold. Note that relying on colour histograms for object recognition clearly limits the types of scenes and object sets we can handle, but as our main goal is on exploring shared grounding, we do not seek absolute competence in vision.

(4) *Spatial region growing*: The resolution of the occupancy grid is then reduced (for efficiency) and next used by a region growing algorithm [24] to group the zones of interest into regions that correspond to objects. The result of this layer is therefore a stream of 'best' hypotheses for each object's coarse spatial occupancy.

(5) *Temporal tracking*: The first four layers all work on streams of single images. The next layers work on multiple images with the goal of tracking the same object over time. Because objects are identified using the histogramming technique, it is relatively easy to know whether they re-occur in the image and to compute their centre so that a trajectory can be established. No sophisticated region tracking (by trying to match parts of regions from one image to the next) is performed, which of course restricts the number of objects of the same colour that can be handled.

Obviously, this vision system has a number of clear limitations. The background has to be learned prior to visual processing and has to be updated when there is a significant change. The frame rate of the camera is low (24 images/s), so that fast actions (such as a ball being dropped onto the table) cannot be detected. Objects which can participate in scenes have to be learned in advance. There is only a small set of objects (typically seven depending on their colour histograms) that can be used simultaneously in the same set of scenes. Nevertheless, this system is adequate enough for our main purposes (namely experiments in grounded language communication). Work continues to improve its reliability and speed by integrating additional vision algorithms, but the strong temporal constraint imposed by the real world puts a limit on how much visual processing can be done with available hardware. Right now the system computes all visual information at a rate that keeps up with the frame rate of the camera.

4.2. Event detection and tracking

The next set of visual processes is concerned with the detection and tracking of events. The task is similar

to that of detecting objects, in the sense that deictic representations are constructed and maintained. The main difference is that grouping is based on changes in the properties of objects rather than on invariances. Event detection is organised in different layers:

(1) *Detecting change*: The first layer produces a stream of properties of objects that change over time. Specifically, it produces qualitative descriptions for:

- Movement of an object, which is signalled if the centre of gravity of an object has changed significantly in between two image frames.
- Contact between two objects which is signalled if the regions of the objects concerned touch each other for a significant time period.
- Approach between objects which is triggered if the distance between the centres of gravity of two objects is becoming significantly smaller between image frames.
- Front positioning of one object x with respect to an object y which is signalled if x is moving towards y, and y is located within a cone emanating from x.

A stream of Boolean values for these descriptions are produced for all the objects in the scene, together with an indication of their certainty.

(2) *Detecting events*: The next layer groups these results in time. Moments when the same configuration of qualitative descriptors holds are grouped together in blocks. For the scenes shown in Fig. 2, two objects are being tracked: 0 (a hand) and 1 (a red object), and the following properties: the hand moves (M0), object-1 moves (M1), object-1 and the hand touch each other (T10), they approach each other (A10), object-1 moves in front of the hand (F10) or the hand moves in front of object-1 (F01). The description streams generated in conjunction with Fig. 2 is as follows, where the number in front of each line indicates the number of time steps that the same configuration of descriptions holds.

```
    [M0  M1  T10 A10 F10 F01]
13 [ 1            1         ]
12 [          1             ]
 7 [ 1   1    1           1 ]
 6 [          1             ]
```

Blocks of time in which the same configuration holds are called micro-events. For example, the first micro-event is one where the hand moves and approaches an object. The second micro-event is one where the hand is touching the object. In the third micro-event the hand and the object move, the hand still touches the object and the hand is moving fronted with respect to the object. In the final micro-event the hand still touches the object but neither the hand nor the object move. Micro-events are generated as soon as they have been found, in other words when the configuration of qualitative descriptors changes for a significantly long time.

(3) The final layer is concerned with the detection of events. Events are sequences of micro-events. For example, the pick-up event in Fig. 2, involving a hand and an object, is defined as a sequence of three micro-events: (1) the hand moves towards the object, (2) the hand touches the object, and (3) both move away together. Processes concerned with recovering such events use a library of event definitions which is matched against the stream of micro-events.

4.3. Qualitative descriptors

The final set of processes consists of pattern detection algorithms which detect significant features about the objects or events at different levels of the hierarchy. These algorithms look at the stream of units and compute properties of single objects (such as size, shape, colour, texture, etc.) or properties of multiple objects (such as respective geographical locations and change in locations). They output the result as streams of qualitative descriptions with a certainty indication. The algorithms use standard techniques from computational geometry and pattern recognition [6].

The qualitative descriptors are integrated in a flexible architecture that makes it possible to add new detectors at any time at any level of the object or event hierarchy or reschedule their usage, partly driven by top-down expectations. Concretely each descriptor runs as a separate parallel process (implemented as POSIX threads). Each process gets time-slices to advance its computation. Certain algorithms require more resources than others, and so ‘quick’ algorithms yield early results which can already be used at higher

levels and may be sufficient for the purpose of language communication. Processes may be pre-empted when their results are no longer relevant.

4.4. Top-down information flow

In the discussion so far we assumed that information flows only in a bottom-up manner: from the images captured by the camera via a whole set of processes to the facts in memory. But this is a simplification that does not work for two reasons:

1. Each of the processing steps discussed may yield an unreliable result. For example, it is seldom the case that the qualitative descriptors which provide the basis for the detection of micro-events yields a clean set of outputs so that the micro-event is neatly defined as a block in time. Instead, the configuration is interspersed with very short moments when some of the descriptors do not hold. If we would only perform strict bottom-up processing we cannot deal with this kind of noise. Our solution has been to introduce for each pattern detector a top-down influence from the next level up. The user of the results of a pattern detectors monitors the certainty of recognition and the constancy of a pattern over time, so that small glitches can be eliminated and weak hypotheses discarded.
2. There is so much visual information in the image that it is impossible to extract fast enough everything that could possibly be extracted. Moreover, as little as possible should be put into the fact memory to avoid overloading or slowing down symbolic processing. However, occasionally more processing at lower levels is necessary: because an object being tracked on the basis of colour disappears temporarily from view, or because the continuation of an action which was taking place in fact does not take place, because the listener needs to use information about the shape of an object which was not yet computed by the vision system, etc. In these situations, it must be possible to assign more resources to the processes taking place at a specific layer and perform additional computation.

We have addressed these issues by introducing the notion of requests. Requests can be sent from the language system to the vision system, and the vision system can internally also generate requests. Requests trigger the activation or re-activation of pattern detectors on specific stretches of the object or event streams. They may also cause the reconfiguration of pattern detectors to change priorities and give more computer time to requested information. Finally, they can change the set of descriptors that is sent by default from the vision system to the fact memory.

Here are two concrete examples where this facility is used:

(1) In deciding what to say, the speaker must find a distinctive description to refer to an object. Suppose that there are two objects in memory, obj-1 and obj-2, and that facts in the fact memory only say that they are both red, perhaps because colour was the only property computed so far with sufficient certainty. The speaker cannot discriminate between the objects and so a request is issued to the vision system to stimulate computation of other qualitative descriptors for obj-1 and obj-2, that might yield a distinctive description. There are pattern detectors for colour, shape, texture, size, position, etc. with default priorities. Some of them may not have had enough resources to come up with a reliable conclusion, others may have such a low priority that they were not started at all. When the request comes, more computational power is given to these pattern detectors. Moreover, they do not necessarily operate over the image stream as it is entering the system but on past stretches as recorded in visual memory. When the pattern detectors produce more results, they are sent to the language system, turned into facts in memory and used in a new attempt for discrimination.

(2) The hearer may be sent an utterance that uses a set of properties which are not yet in the fact memory. For example, the hearer may have been asked to identify “the ball next to the green cube” but his fact memory may have recorded only that there is a green and a red object. Again a request is generated to the hearer’s vision system to go after more information. This request can be precise: compute information about shape for these specific objects given the hypothesis that it can be ball or a cube.

It would be desirable to enrich the power of the top-down flow of constraints on vision processing, for example, by predicting the position of objects in future time steps and use that as hypotheses for the pattern

detectors, but this is not done yet in the current implementation.

5. Design principles for grounding

We now attempt to extract some of the lessons learned from our designs and experimentations with the grounded communication system briefly described in the previous section. We do not claim that these principles are unique to the system discussed here, on the contrary, we try to capture the 'best practice' in the field.

1. *Indexical representations*: The first important principle is to introduce a continuous detection and tracking of objects and events. The vision system described here latches onto an object or event and keeps tracking it as much as possible. This results in a dynamic deictic representation which maintains streams of indexical references, even if objects or events change.
2. *Description streams*: The second important principle is the introduction of description streams which produce and monitor properties of objects and events in time. The streams start from the images flowing in through the camera at a steady rate and continues all the way up to facts spilling into the fact memory. The units to which the descriptions apply are assembled, first spatially then temporally, at many different hierarchical levels.
3. *Noise reduction*: It is well known that real images taken from relatively unprepared real world situations always yield noisy processing results. This motivates the next design principle: noise at one hierarchical level can be reduced by preferring the most coherent analysis at the level just above it. We apply this principle through the whole system and at all levels.
4. *Top-down information flow*: It is clearly not enough to have information flowing from the sensory data to the conceptual world model, partly because there is not enough time to compute everything that could be computed. So there must also be a steady top-down flow of requests and expectations from the 'cognitive' layers to sensory processing.
5. *Attention*: Finally, we believe that an attention mechanism is unavoidable. The attention mechanism is responsible for allocating scarce computation resources. The system discussed in this paper uses a variety of means to achieve this: figure/ground computation at a very early stage, a thread-based implementation of feature detectors with varying and dynamically modifiable priorities partly steered by the vision system.

6. Conclusions

Grounded robots that engage in communication using external representations not only need a physical body and low-level behaviours but also a conceptual world model which must be anchored firmly and dynamically by the robot in the environment through its sensori-motor apparatus. We argued that there is not a simple sweeping theoretical principle to turn a system that uses conceptual world models into a grounded system. Instead many processes must be carefully integrated. We described an implemented system that has attempted to do so in the context of experiments in grounded open-ended language communication among robots as well as between humans and robots. We also proposed a set of design principles that capture the principles that we have used in our design.

Acknowledgements

This research was done at the Sony Computer Science Laboratory in Paris. We are grateful to several members of the laboratory for feedback and comments on the work discussed here. The research is also part of a French CNRS OHLL research project on the Origins of Language.

References

[1] P. Agre, D. Chapman, Pengi: An implementation of a theory of activity, in: Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI Press, Anaheim, CA, 1987, pp. 268–272.
[2] J.F. Allen, G. Ferguson, Actions and events in interval temporal logic, Journal of Logic and Computation 4 (5) (1994) 531–579.
[3] I. Androutsopoulos, G.D. Ritchie, P. Thanisch, Natural Language Interfaces to Databases—An Introduction,

Cambridge University Press, Cambridge, Journal of Natural Language Engineering 1 (1) (1995).
[4] N. Badler, M. Palmer, R. Bindiganavale, Animation control for real-time virtual humans, Communications of the ACM 42 (8) (1999) 64–73.
[5] J.-C. Baillie, J.-G. Ganascia, Action categorization from video sequences, in: W. Horn (Ed.), Proceedings of the ECAI 2000, IOS Press, Amsterdam, 2000.
[6] J.-C. Baillie, Apprentissage et reconnaissance d'actions dans des sequences video, Ph.D. Thesis, Universite de Paris.
[7] J.-C. Baillie, Object tracking using certainty color maps, in: Proceedings of the ECCV'2002, submitted for publication.
[8] R.A. Brooks, Intelligence Without Representation: Artificial Intelligence Journal 47 (1991) 139–159.
[9] H.H. Clark, S.A. Brennan, Grounding in communication, in: L.B. Resnick, J.M. Levine, S.D. Teasley (Eds.), Perspectives on Socially Shared Cognition, APA Books, Washington, 1991.
[10] P. Cohen, et al., Proceedings of the AAAI Spring Symposium on Grounding, AAAI Press, Anaheim, CA, 2001.
[11] S. Coradeschi, D. Driankov, L. Karlsson, A. Saffiotti, Fuzzy anchoring, in: Proceedings of the IEEE International Conference on Fuzzy Systems, Melbourne, Australia, 2001.
[12] A. Elfes, Using occupancy grids for mobile robot perception and navigation, Computer 22 (6) (1989) 46–57.
[13] S. Harnad, The symbol grounding problem, Physica D 42 (1990) 335–346.
[14] Y. Kuniyoshi, H. Inoue, Qualitative recognition of ongoing human action sequences, in: Proc. IJCAI-93, 1993, pp. 1600–1609.
[15] N. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, Los Altos, CA, March 1998.
[16] L. Shapiro, G. Stockman, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 2001.
[17] J. Siskind, Visual event classification through force dynamics, in: Proceedings of the AAAI Conference 2000, AAAI Press, Anaheim CA, 2000, pp. 159–155.
[18] D. Sperber, D. Wilson, Relevance: Communication and Cognition, Harvard University Press, Cambridge, MA, 1986.
[19] L. Steels, Language games for autonomous robots, IEEE Intelligent Systems (2001) 16–22.
[20] L. Steels, Simulating the origins and evolution of a grammar of case, in: Proceedings of the Evolution of Language Conference, Harvard, April 2002, in press.
[21] L. Steels, F. Kaplan, AIBO's first words, The social learning of language and meaning, Evolution of Communication 4 (1) (2001).
[22] L. Steels, F. Kaplan, A. McIntyre, J. Van Looveren, Crucial factors in the origins of word-meaning, in: A. Wray (Ed.), The Transition to Language, Oxford University Press, Oxford, UK, 2002.
[23] L. Talmy, Toward a Cognitive Semantics: Concept Structuring Systems (Language, Speech, and Communication), The MIT Press, Cambridge, MA, 2000.
[24] M.G. Montoya, C. Gil, I. Garcia, Implementation of a region growing algorithm on multicomputers: analysis of the work load balance, in: Proceedings of the AI'2000, Canada, 2000.

Luc Steels is professor of Artificial Intelligence at the University of Brussels (VUB) and director of the Sony Computer Science Laboratory in Paris. He is one of the pioneers of artificial life approaches to robotics, particularly in the domain of behavior-based architectures. More recently he has developed evolutionary approaches to language, in which robots evolve communication systems through self-organisation and emergence. He published a dozen books in several areas of Artificial Intelligence.

Jean-Christophe Baillie studied physics and computer science at the Ecole Polytechnique, France, and holds a Ph.D. thesis on Artificial Vision Computer Science from the Computer Science Laboratory of Paris 6. He conducted the research for this paper at the Sony Computer Science Laboratory in Paris. At the moment he is Associate Professor at ENSTA (Ecole Nationale Supérieure des Techniques Avancées).