

# Language Games with Mixed Populations

Michael Lewin and Emmet Spier

School of Cognitive and Computing Sciences,  
University of Sussex, Brighton, UK  
mlewin@bigfoot.com

**Abstract.** This paper presents an adaptation of Luc Steels’s model of Category Formation and Language Sharing. The simple competitive learning algorithm is proposed as a more general means of creating categories from real-world perception. The model is shown to achieve high levels of coherence and to be very robust when two distinct populations are mixed together, with both populations learning each other’s words.

## 1 Introduction

Luc Steels has established a model of concept sharing in artificial agents using two mechanisms: firstly, the partitioning of a perceptual Input Space as a means of *Category Formation*, and secondly, the use of Language Games amongst agents as a means of *Language Sharing*. The result is a population of agents with a high degree of coherence in their use of language. See [15] for an overview of Steels’s “Talking Heads” experiment, which utilises these two mechanisms.

This paper is an extension of Steels’s work (especially [9], [10] and [11]) which attempts to create a more general model of category learning. The protocol of the Language Games [9] is largely unchanged, but the nature of the perceptual input and the subsequent method of Category Formation is very different. *Simple competitive learning* [4] is used to partition the Input Spaces, which are vector spaces and can be of any dimension. This increased generality should enable Steels’s model to be applied successfully to a wide variety of practical situations.

The use of *symbols* has been a source of much debate in Cognitive Science. Newell and Simon’s *physical symbol system hypothesis* [6] – that physical symbol systems are necessary and sufficient conditions for intelligence – has been challenged, in particular by Searle’s Chinese Room argument [8] and behavioural-based AI approaches (e.g. Brooks [1]). In response to Searle, Harnad [3] suggested that the only way to escape the “symbol/symbol merry-go-round” of empty syntax is for a set of elementary symbols to be grounded through perception in the real world. All other symbols can then be derived from this elementary set. This *symbol grounding problem* is an important theme in Steels’s work, which addresses the need to bridge the gap between perception and his concepts by partitioning a continuous perceptual Input Space into a finite set of discrete categories.

Brooks [1] proposed that we can do away with symbols altogether, but his claim can only be justified by empirical evidence. Thus far, non-symbolic approaches have been effective in models of low-level tasks, but intuition suggests

that symbols are still useful in talking about – and perhaps even essential in dealing with – higher level tasks, in particular language.

As a means of compromise in the symbol debate, and possibly of bridging the gap between low- and high-level tasks, Vogt [16] suggests that Pierce’s definition [7] of symbol is adopted<sup>1</sup>. A *semiotic symbol* is defined as the relationship between a referent (e.g. an object), its meaning, and an arbitrary or conventionalised form (e.g. a word). It is difficult to describe exactly what “meaning” is, but Vogt says it “can be viewed as a functional relation between a form and a referent”.

Category Formation is central to our model, so how does this relate to the semiotic symbol? According to Vogt, “although a category should not be equated with a meaning, it is labeled as such when used in communication, because it forms the memorized representation of a semiotic symbol’s meaning”. The similarity between “meaning” and “concept” is exposed here – a “category” can be seen as an elementary form of “concept”, where the concept is defined only in terms of its members rather than by some kind of meta-description of the category.

The paper is laid out as follows: Section 2 gives an introduction to Steels’s framework and explains how our model differs from Steels’s. Details of the model are given in Sections 3 and 4. Steels’s work is simulated in Section 5, and the importance of a *forgetting mechanism* is highlighted. The robustness of the model when two different populations are mixed together is demonstrated in Section 6.

## 2 Extending Steels’s Model

The key difference between the proposed model and Steels’s is the nature of the Input Spaces and the categories formed within them. In Steels’s model, an Input Space is taken to be the one-dimensional bounded real segment  $[0,1]$ . Initially the entire space constitutes one category. New categories are formed as a consequence of an unsuccessful Language Game; an existing category is bisected to form two new ones in the hope of creating a distinction between two previously indistinguishable objects.

In our model, an Input Space is taken to be the  $n$ -dimensional vector space  $\mathbb{R}^n$ . In contrast to Steels’s model, these Input Spaces may be of any dimension and take arbitrarily large positive or negative real values. Categories are formed using a simple competitive learning algorithm [4], an example of unsupervised learning. It partitions the space according to the naturally existing clusters of vectors<sup>2</sup>. The input vectors could be defined by a statistical distribution or sampled from a real-world situation.

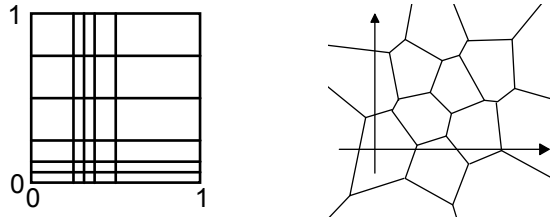
Steels’s model can account for higher dimensional spaces by combining several one-dimensional spaces which are considered to be orthogonal. This leads to

---

<sup>1</sup> Steels also advocates Pierce’s terminology in [15].

<sup>2</sup> It must be assumed that the input is presented in such a way that members of the same category are indeed clustered together. This is what Harnad calls the “Categorical Perception (CP) Effect” [2], but it will not be discussed further here.

a partitioning of the space that is less natural and more restricted than the competitive learning algorithm's, as illustrated in Fig. 1.



**Fig. 1.** The difference between Steels's model (left) and the new model (right) is the nature of the Input Spaces and the categories formed within them.

In Steels's model, the agents associate words with *feature sets*, where a *feature* is defined as an *attribute-value pair*. For example, a one-dimensional attribute such as "horizontal position" could be divided into values such as "far left", "left", "centre" and "right". In our model, each *dimension* of the Input Space will usually correspond to an *attribute*. When the Input Space is one-dimensional, the result is similar to Steels's model – a feature is a region in a single dimension.

In some cases, one may wish to model an attribute as multi-dimensional. For example, colour can be thought of as a three-dimensional space whereby every colour is defined by its red, green and blue components. In such a case it is natural to treat the Input Space as three-dimensional and give names to the regions of this space, without allowing the agent direct access to the three underlying dimensions. The categories will correspond not to regions in a single dimension but to regions in the higher dimensional space. They can still be thought of as *attribute-value* pairs, for example "the colour purple", where "colour" is the attribute and "purple" its value (a region in  $\mathbb{R}^3$ ).

It is also possible, however, to take a single high-dimensional Input Space to be the entire space of possibilities of objects in the world. Thus agents would create names for objects (e.g. "tree") as opposed to attributes (e.g. "big") and the lower-lying attributes of an object (as defined by the dimensions of the space) would not be directly accessible to the agent. It is a question of design whether the agents will create words which correspond to objects, attributes or both.

In this paper, the agents are simulated and their perceptual stimuli are abstract (clusters of points in a vector space). There would be no difficulty, however, in transferring the model to a physical situation such as that which Steels has presented [15].

### 3 Simple Competitive Learning

Our model uses the standard winner-takes-all algorithm with leaky learning. The network consists of a fixed number of *nodes* and with each node is associated

a vector of the same dimension as the Input Space. Initially the vectors are randomly distributed according to the uniform distribution  $U[-2,2]$ . Whenever the network is presented with an input vector, the *winning node* is the one whose corresponding vector is closest (in Euclidean distance) to it. The winning vector is then updated according to the following rule:

$$d\mathbf{w}_i = \eta(\mathbf{I} - \mathbf{w}_i) \quad (1)$$

where  $\mathbf{I}$  is the input vector and  $\mathbf{w}_i$  is the weight vector corresponding to the winning node  $i$ . The learning rate  $\eta$  was always set to 0.1.

The winning vector, which was already closest to the input vector, moves closer still. This is the positive feedback mechanism which enables the prototype vectors to identify clusters in the Input Space. This alone, however, will not always be successful in identifying the clusters. There will often be *dead nodes* which are never the winner and hence never move. To remedy this, it is common practice to add *leaky learning* to the model; the winner is updated according to (1) and all the other vectors are updated by the same rule but with a smaller learning rate. We used a rate of  $\frac{\eta}{100}$ .

This leads to a very stable situation. The prototype vectors are able to locate the clusters quite quickly and, more importantly, once the clusters have been located the prototype vectors stay in the region of that cluster. Thus there is no need to reduce the learning rate after training, as is common with other neural network algorithms.

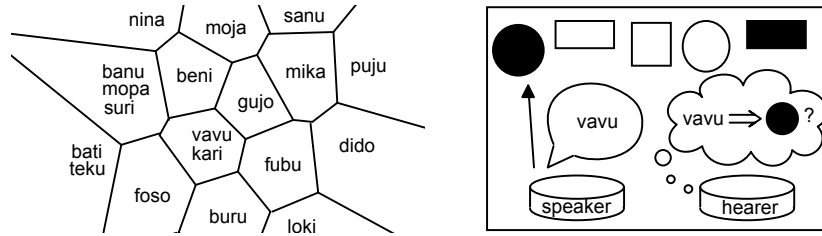
At any time the Input Space can be thought of as being partitioned by the “*Voronoi Sets*” [4] defined by the prototype vectors. Each such set, or *proximity neighbourhood*, contains precisely one prototype vector and is defined as the set of points which are closer to that prototype vector than any other.

## 4 Language Games

Once the necessary categories have been formed, the question arises of how a whole society of agents can share their categories using language, i.e. a set of words with commonly agreed meanings. In a realistic setting, it is important to consider language in the context of some *selective pressure*: language will only arise because it is beneficial in some way to either the individual or the species. Steels [12] [13], however, argues that the contrived notion of a *Language Game* is a useful tool for rigorously analysing the process of language acquisition. Here, a Game is defined as “a routinised sequence of interactions between two agents involving a shared situation in the world”. In this case it involves a speaker trying to describe an object to a hearer by uttering words which refer to the object’s features. An overview of our model is shown in Fig. 2.

The protocol of the Language Games is almost identical to that used by Steels. Two agents are selected at random to play the roles of *speaker* and *hearer* respectively. A fixed number of objects are created<sup>3</sup> to form the *context* and one

<sup>3</sup> In Steels’s model the objects are chosen from a finite (and small) collection. In our model, they are created afresh each time, allowing for an infinite variety of objects.



**Fig. 2.** Overview of the model. Words can be associated with the Voronoi Sets formed by competitive learning (left). The agents achieve a consensus through interacting in Language Games (right).

of them is designated (as if the speaker had pointed at it) to be the *subject* which the speaker will try to describe.

The speaker then generates a *distinctive feature set (DFS)* – a set of features which the topic possesses and which none of the other objects in the context has. If more than one DFS exists, the smallest is chosen. When more than one minimal DFS exists, preference is given to feature sets for which the agent already has a word. To choose between two such feature sets, the entry’s “*score*” (*successes – failures*) is used as a tie-breaker – the entry with the highest score is selected. This is the crucial positive feedback mechanism which drives the system towards coherence – agents prefer words which are already established and successfully used in communication. If no DFS exists, the Language Game is aborted.

The speaker converts its DFS to a word<sup>4</sup> using the *cover* function.<sup>5</sup> This searches the lexicon for an occurrence of the feature set and returns the corresponding word. If more than one such occurrence exists, the entry’s *score* (*successes – failures*) is again used as a tie-breaker. If no such occurrence exists, it is possible (with probability  $p_w = 0.05$ ) for a new word corresponding to the DFS to be created and stored in the lexicon. If this does not occur, the game is aborted. The parameter  $p_w$  affects how many different words enter the population, for the lower it is, the more chance there is for a single word to be spread around the population instead of many words with the same meaning being invented by different agents.

The hearer then converts the word back to a feature set using the *uncover* function. This simply searches the lexicon for that word. If the word does not appear in the hearer’s lexicon, its meaning is guessed – a new entry is created comprising the uttered word and a DFS selected by the hearer. It is possible that this is not the same DFS selected by the speaker, so ambiguities can arise.

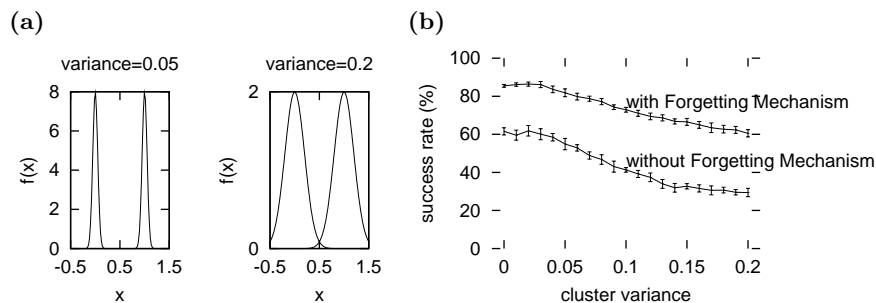
<sup>4</sup> all “words” have four letters of the form consonant, vowel, consonant, vowel. Of course any arbitrary symbols could be used as “words”.

<sup>5</sup> The *cover* and *uncover* functions defined here are simpler than Steels’s [9]. In particular, *utterances* (sets of more than one word) were deemed unnecessary.

If the uttered word was not new to either the speaker or hearer then the game is evaluated (as if the hearer had nodded or shaken its head in response). The hearer interpreted the word as a feature set. If this is a DFS for the subject, the game is recorded as a success. If not, it is recorded as a failure. This feedback will affect the scores of the word/category pairs in the agents’ lexicons.

## 5 Simulation of Steels’s Model

Initially, a situation similar to that proposed by Steels in [10] was considered. A population of 30 agents was trained for 10,000 input steps before the Language Games commence to allow the competitive learning algorithm ample time to establish stable categories. A series of 200,000 training Language Games was then carried out. Each Language Game involved 5 objects. Every object comprised 5 parts, each drawn from a one-dimensional Input Space. The five Input Spaces all had the same distribution: just two clusters distributed normally with means at 0 and 1 respectively and identical variance  $\sigma^2$ . This is illustrated in Fig. 3a.



**Fig. 3.** (a) Points in the Input Space are defined by two normal distributions centred at 0 and 1. The variance  $\sigma^2$  affects how much overlap there is between the two clusters. (b) Long-Term Success Rate against cluster variance  $\sigma^2$ , with and without forgetting mechanism. Each bar shows the mean and one standard deviation over 10 runs.

When  $\sigma^2$  is very small, the agents’ categories are almost fixed – only the “leaky learning” mechanism causes a slight plasticity in the prototype vectors’ positions. This is comparable to Steels’s model. As  $\sigma^2$  increases, agents’ categories become less stable and so different agents will have different perceptual categories, leading to a decrease in success of the Language Games.

A useful way to quantify this is to define the *Long-Term Success Rate*  $\rho$  as

$$\rho = \left( 100 \times \frac{\text{successes}}{\text{successes} + \text{failures}} \right) \quad (2)$$

where the successes and failures, as defined in Section 4, are recorded over 1,000 test Language Games after the 200,000 training Language Games have been carried out.

In later work [15] Steels adds a *forgetting mechanism* to his model. This allows word/category pairs to be removed from an agent’s lexicon if they have not been used for a long time (20,000 rounds), or if their score *successes* – *failures* falls below a given threshold (0).

The forgetting mechanism significantly increases the communicative success of the population. Without it, any word/category pair recorded in an agent’s lexicon remains fixed for the duration of the experiment. Thus if there is a single case of misunderstanding between two agents, the erroneous word/category association will lead to repeated failures in the Language Games.

## Results

Figure 3b shows the Long-Term Success Rate  $\rho$  with and without the forgetting mechanism. It is clear that  $\rho$  is significantly higher when the forgetting mechanism is used, achieving highs of almost 90% for small  $\sigma^2$ . By comparison, without the forgetting mechanism,  $\rho$  is never more than 65%. In both cases,  $\rho$  decreases as  $\sigma^2$  increases, as expected<sup>6</sup>. When  $\sigma^2 = 0$  these results replicate Steels’s.

## 6 Mixing Two Populations

An agreed set of linguistic conventions can be tested by allowing two populations to develop independently before mixing and engaging in Language Games together. The agents will already have established their own linguistic conventions, but they should also be able to learn the new words of the foreign population. This can be considered a crude model of the meeting of two different nationalities with entirely different words for the same meanings.

Steels [14] demonstrated that under his model, a linguistically naive agent can be added to the population and will successfully learn the linguistic conventions present in the society. Here, two mature populations are mixed together – a much sterner test of stability.

Two populations of equal size (30 agents)<sup>7</sup> were trained for 10,000 rounds and then engaged in training Language Games for 200,000 rounds using identical parameter values to those in Section 5. Then the agents in the second population were pooled with a set of *visitors* from the first population to create a mixed population. A further series of training Language Games was then carried out in which the speaker and hearer were drawn at random and with equal probability from this mixed population. Thus it was possible that the speaker and hearer came from different populations, but there would still be interactions between

---

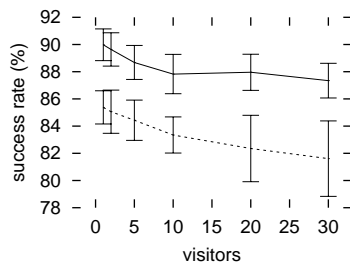
<sup>6</sup> The model was also effective with higher dimensional Input Spaces. Increasing the number of dimensions does not directly affect  $\rho$ , but other related parameters, such as the number of categories formed by the agents and the degree to which data is clustered together, will. Details can be found in [5].

<sup>7</sup> If these two populations were not of equal size then the agents from the smaller population would engage in more Language Games. Thus their words would have a higher score than those of the other population, causing them to dominate.

agents from the same population as well. This phase was continued for another 200,000 rounds in order to investigate the long-term effect of mixing populations. The effect of varying the number  $\nu$  of visitors from 1 to 30 was investigated.

## Results

Figure 4 shows two different measurements of success plotted against  $\nu$ . Each bar shows the mean and one standard deviation over 40 recorded values of the success rates. The upper set shows the Long-Term Success Rate  $\rho$  (as defined in Section 5) measured over 1,000 test Language Games after the mixed population has engaged in 200,000 training Language Games. The lower set shows a Short-Term Success Rate  $\rho_*$  measured over the first 1,000 training Language Games immediately after mixing.

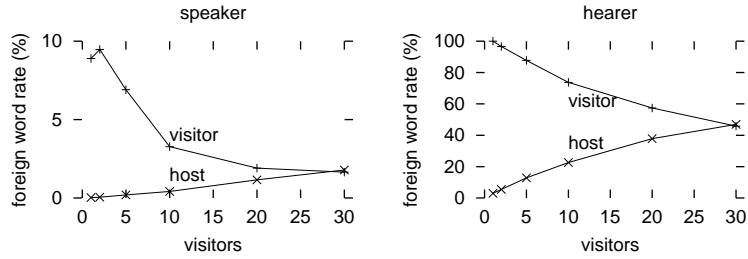


**Fig. 4.** Long-Term (solid) and Short-Term (dashed) Success Rates after mixing populations for varying number  $\nu$  of visitors. Cluster variance  $\sigma^2 = 0.001$ .

The graph demonstrates that the model is very robust when two populations are mixed. Even the Short-Term Success Rate is very high, around 85% for small  $\nu$ , indicating that the agents are able to take on the new words very quickly without a major drop in communicative success. For a fixed value of  $\nu$ , there is an improvement of about 5% from the Short- to the Long-Term Success Rate. The mixed population is able to recover from the initial drop in communicative success - in fact, it can attain success rates as high as would have been seen had the mixing not occurred (not shown). Both rates appear to fall with  $\nu$ , but there is quite a large variance over the 40 runs. For a given value of  $\nu$ , the variance of  $\rho_*$  is greater than that of  $\rho$ .

In order to quantify the amount of lexical exchange taking place, it is useful to define the *Foreign Word Rate* - the proportion of times that a word spoken or heard by an agent comes from the population to which it does not belong. Figure 5 shows this rate in four different situations - spoken or heard, and by an agent from the visiting or host population - for different values of  $\nu$ .

Words are foreign to the hearer much more often than they are to the speaker because agents tend to prefer to utter their own words. That foreign words are ever spoken, however, implies a successful integration of the two populations.



**Fig. 5.** Foreign Word Rates after 200,000 rounds for mixed populations. Each population comprised 30 agents but the number of visitors is varied. On the left, visiting and host speakers are compared. On the right, visiting and host hearers are compared.

As  $\nu$  increases, the hosts speak or hear foreign words more frequently while the visitors do so less. In fact the ratio between the visitors’ foreign word rate and the hosts’ is the inverse of the ratio between the number  $\nu$  of visitors and the 30 hosts. This can be explained as follows: the population ratio governs the proportion of times a hearer will encounter a foreign speaker and (probably) hear a foreign word. Successful interactions of this kind will increase the score of the foreign word and, in time, the foreign words can become the preferred word for a particular feature set and will be uttered when that agent is the speaker.

## 7 Discussion

This paper has demonstrated that an unsupervised learning technique, simple competitive learning, can be successfully coupled with Luc Steels’s model of Language Games. High levels of coherence were achieved, although the presence of a “forgetting mechanism” was crucial for this.

The new technique used in the Category Formation stage has some important similarities to that used in the Language Sharing stage. Both achieve coherence through a *positive feedback* mechanism, namely the winner-takes-all rule and the preference for successful words respectively. Also, although an initial training period is necessary for the agents’ networks to stabilise, there is no discontinuity between this stage and the subsequent stage of Language Games. At all times, the agent’s perceptual apparatus is presented with objects from the environment, and the agent’s internal network is updated according to the competitive learning algorithm. The result is a more unified overall framework.

Steels believes that the interaction between language and conceptualisation is very important, and his model achieves this since new categories were formed as an outcome of the Language Games. In our model, there is much less interaction because the number of categories is a fixed parameter of the agent’s network, but the model is stable nonetheless. It is not practical to create arbitrarily fine distinctions between objects as Steels proposes; sometimes, two objects really are perceptually identical. Furthermore, the generalisation through prototypes

is a more robust model in a noisy environment because in Steels's model fine distinctions would be created between identical objects as an artifact of the noise.

The lexicons' robustness has been demonstrated through the agents' ability to adapt to new linguistic conventions without sacrificing their existing ones. Even when the visiting population is small, foreign words can be adopted and spread around a population of agents without any significant fall in coherence, something commonly observed in human language.

## References

1. Brooks R. A.: Intelligence without representation. *Artificial Intelligence* **47** (1991) 139–159
2. Harnad, S.: Category induction and representation. In: Harnad, S. (Ed), *Categorical perception: The groundwork of cognition* (1987)
3. Harnad, S.: The symbol grounding problem. *Physica D* **42** (1990) 335–346
4. Hertz, J., Krogh, A., Palmer, R.: *An Introduction to the Theory of Neural Computing*. Pub. Addison-Wesley CA (1991)
5. Lewin, M.: *Concept Formation and Language Sharing: Combining Steels' Language Games with Simple Competitive Learning*. Masters thesis, School of Cognitive and Computing Sciences, University of Sussex (2002). Also available at [http://www.cogs.susx.ac.uk/lab/adapt/EASy\\_MSc\\_abs\\_02.html](http://www.cogs.susx.ac.uk/lab/adapt/EASy_MSc_abs_02.html)
6. Newell, A., Simon, H. A.: *Computer Science as empirical enquiry: Symbols and search*. *Communications of the ACM* **19** (1976) 113–126
7. Pierce, C. S. *Collected papers, vol I–VIII*. Cambridge MA: Harvard University Press (1931–1958)
8. Searle, J.: *Minds, Brains, and Programs*. *Behavioral & Brain Sciences* **3** (1980) 417–458
9. Steels, L.: *Emergent Adaptive Lexicons*. In: Maes, P., Mataric, M.J., Meyer, J.-A., Pollack, J., Wilson, S.W. (Eds), *From animals to animats 4: proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* (1996) 562–567
10. Steels, L.: *Constructing and Sharing Perceptual Distinctions*. In: van Someren, M.W., Widmer, G., *9th European Conference on Machine Learning* (1997) 4–13
11. Steels, L.: *Synthesising the Origins of Language and Meaning Using Co-Evolution, Self-Organisation and Level Formation*. In: Hurford, J.R., Studdert-Kennedy, M., Knight, C. (Eds), *Approaches to the Evolution of Language: social and cognitive bases* (1998) 384–404
12. Steels, L.: *Language Games for Autonomous Robots*. *IEEE Intelligent Systems* **16(5)** (2001) 16–22
13. Steels, L., Kaplan, F.: *AIBO's first words. The social learning of language and meaning*. *Evolution of Communication* **4(1)** (2001)
14. Steels, L., Kaplan, F., McIntyre, A., Van Looveren, J.: *Crucial Factors in the Origins of Word-Meaning*. In: Wray, A. (Ed), *The Transition to Language* (2002) 214–217
15. Steels, L.: *Grounding Symbols Through Evolutionary Language Games*. In: Cangelosi, A., Parisi, D. (Eds), *Simulating the Evolution of Language* (2002) 211–226
16. Vogt P.: *The physical symbol grounding problem*. *Cognitive Systems Research* **3** (2002) 429–457