Lecture Notes in Artificial Intelligence, Vol.5040, pp.331-341, 2008
Proc. 10th Int. Conf. on Simulation of Adaptive Behavior (SAB2008), Osaka, Japan.

Acquiring a Functionally Compositional System of Goal-directed Actions of a Simulated Agent

Yuuya SUGITA and Jun TANI

RIKEN Brain Science Institute, Hirosawa 2-1 Wako-shi 3510198, JAPAN,
{sugita, tani}@bdc.brain.riken.go.jp

Abstract. We propose a sub-symbolic connectionist model in which a functionally compositional system self-organizes by learning a provided set of goal-directed actions. This approach is compatible with an idea taken from usage-based accounts of the developmental learning of language, especially one theory of infants' acquisition process of symbols. The presented model potentially explains a possible continuous process underlying the transitions from rote knowledge to systematized knowledge by drawing an analogy to the formation process of a geometric regular arrangement of points. Based on the experimental results, the essential underlying process is discussed.

1 Introduction

In this study, we try to examine the mechanisms in our mind that are involved in the shift from unrelated rote knowledge acquired by learning examples of objects or events into a flexible conceptual system by which we can conceive something not experienced as a recombination of the examples. Skinner (1957) argued that a reusable unit emerges as a by-product of the acquisition of multiple examples containing the reusable unit. He pointed out that a minimal unit seldom appears by itself as whole examples of stimuli and responses. Tomasello (2003) reported that infants can appropriately use holophrases, which are indivisible sentences such as "lemme-see," in a communicative context before understanding the reusable units, which include phrases and words such as "let," "me" and "see." In his usage-based accounts of language development, each transition of the performance is explained in terms of the acquisition of a new type of smaller and more abstract symbolic device.

It is, however, difficult to transfer the idea that the utilization of wholes precedes the emergence of parts from explanatory to computational models. One of the most substantial problems is how to implement the acquisition of a composition rule, which combines smaller units into a whole concept. Cognitive theories often neglect the composition rule, since the rule and the units are considered to be two sides of the same coin (Goldberg, 1995). This belief is plausible only as far as symbolic manipulation is concerned. However, the realization of an embodied composition rule, which is the correspondence of the symbolic manipulation in reality, requires much more than the acquisition of a mere syntactic structure of the symbolic system. Let us consider the case in which an agent generates an action specified by a target object and an operation on it. It is quite easy to represent the action in a symbolic system; a pair of symbols representing the target and operation is enough. On the contrary, an embodied composition rule, which is required to generate an action relevant to the pair, is not so trivial. The problem is that the rule tends to be too abstract to be learned by examples, because the

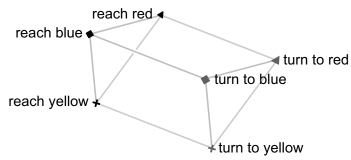


Fig. 1. Systematic relationships among concepts can be represented based on the regularity of a geometric structure.

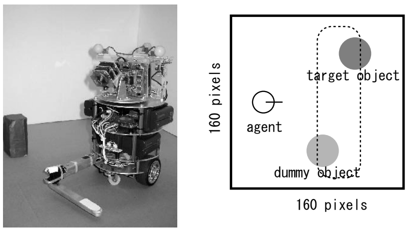
rule needs to capture the anything residual that cannot be collected as symbols, which are usually grounded to something concrete. In fact, many computational models employ a pre-programmed composition mechanism, although their objectives are different from ours (Iwahashi, 2006; Roy, 2002; Cangelosi & Riga, 2006).

In order to avoid the difficulty concerning the abstractness of the explicit composition rule, this study investigates a novel embodied implementation of a functionally compositional system in the domain of goal-directed actions of a simulated agent. A functionally compositional system is one which does not keep any reusable units explicitly in the form of symbols but works like a conventional compositional system (van Gelder, 1990). In other words, there are no symbols, and therefore no composition rules to manipulate them. Instead of dealing with reusable parts explicitly, the functionally compositional system focuses on the systematic relationships among wholes. Each whole concept is embedded as points in a conceptual space implemented as an n-dimensional vector space. The geometric arrangement of these points represents the underlying combinatoriality among them. For example, a system of six actions specified by every possible combination of one of three objects and one of two operations is represented as a triangular prism, as shown in Fig. 1. Even if the positions of some actions are unknown, they can be inferred by utilizing the geometric regularity. Furthermore, this framework explains the transitions from rote knowledge to systematic knowledge in terms of a continuous internal process. The emergence of the regularity involved in the transition can be realized by the continuous motion of each point. It is also remarkable that each whole concept does not change through the transition. Only their relationships are altered, whereas the conventional implementation undergoes the replacement of a holistic symbol with a combination of elemental symbols. Thus, our approach might provide a dynamical interpretation of conventional usage-based models.

In the following, we propose a computational model whereby the geometric regularity self-organizes through the learning of examples. Our experimental setting and connectionist architecture is explained in Section 2. In Section 3, the experimental results, which demonstrate that three different types of combinatorial generalizations are realized by the same model, are presented. An analysis of the result is shown in Section 4, and a possible underlying dynamical mechanism of the functional compositionality is explained in Section 5.

2 Experimental Setting

In our experiment, a simulated mobile agent learns incomplete parts of a total of 36 different goal-directed actions; the actions are characterized by combinations of a target object, an operation on the target, and an optional verb modifier. The learning is conducted under the supervision of teaching programs written for this study. The agent is a model of the mobile robot depicted in Fig. 2(a). This robot has a color camera with



(a) (b)

Fig. 2. The simulated agent. The agent is based on (a) a mobile robot that performs 36 types of goal-directed actions on a stage (b) where a target and an optional dummy colored object are placed randomly within a dashed square.

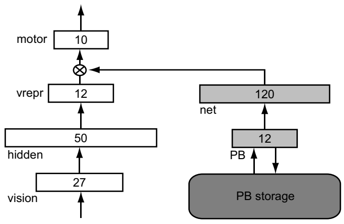


Fig. 3. The architecture of the learning network is presented. A rectangle represents a layer. The number of nodes contained in the layer is denoted by the number in the rectangle. The gray layers have a slower time constant than do the white layers. PB (parametric bias) storage is a working area.

a range of view of 120 degrees and two rotating motors driving each of its two wheels. In each experimental trial, the agent was required to perform either of two operations reach or turnto one of six colored objects (blue, cyan, green, yellow, orange, and magenta) in the environment shown in Fig. 2(b), where one or two objects, one of which is the target, are randomly placed. In actions involving reach, the agent is required to move toward the target and then to stop just before touching it. In the turnto actions, it has to pivot to the target. In our experiment, an operation turnto takes a verb modifier, which designates the offset angles (-30, -18, 0, +18, and +30) of the final position of the target in the visual field from the agent's center. A negative offset indicates the offset to the left, and zero is omitted. In the following, the action is denoted as a triplet consisting of the operation, target, and offset, for example, turnto-blue+18. It should be noted that this notation is used only for our convenience and the agent has no way to access it. In some situations, turnto with an offset is regarded as an operation, and, for example, is denoted as turnto+18.

As mentioned above, the actions are embedded in the concept vector space through the learning process. Unlike the conventional associative learning between an action and a vector, the vector is not provided *a priori*. Instead, the geometric arrangement of the vectors self-organizes the structure, reflecting the relationships among the actions, including unseen ones. The learning model which acquires this structure-preserving map is a connectionist network, as shown in Fig. 3. The network consists of two parts, each of which is tailored to its own functions. One part is a base-level network (base-net) which takes the visual information from the camera as input and outputs the angular velocities of the wheels (the left-hand side of Fig. 3). The base-net is basically a conventional layered neural network except that it has second-order connections (Pollack, 1991) between the vrepr and motor layers. This special mechanism enables the base-net to switch its function. Depending on an action to be performed, the base-net generates different motor values for identical vision input. The second-order connection is controlled by the meta-level network (meta-net) depicted on the right-hand side of Fig. 3. The meta-net is also a conventional layered network. As input, it takes a vector encoding an action once at the beginning of the action, then, it outputs the weights of the second-order connections constantly until the action finishes. The input layer works just like a conventional parametric bias (PB) layer (Tani, Ito, & Sugita, 2004) which has an infinitely long time constant, and therefore we name this as the PB layer. PB storage is as the working area during the learning, as will be explained later in detail. It also keeps the self-organized PB vectors for a later test phase.

An experimental session consists of three phases: the creation of training data, the learning of the data, and the evaluation of the performance. The training data were created by sampling sensor-motor time series involving actions generated in an algorithmic manner. Then, the network learned a part of the data in an offline manner. Four sessions were conducted with supervised data consisting of a different number of actions: 4 and 21 out of 36 actions in the most sparse and dense cases, respectively. After the training error of the network decreased sufficiently, the performance was evaluated. A PB vector of an unseen action was computed by recognizing unused training data, as explained below. The agent, which was controlled by the network, was tested to determine whether it could make a previously unexperienced action in a novel environment with using the PB vector. In the remaining sections, each phase is explained in detail.

2.1 Phase 1: Generating Examples by Teaching Programs

For each of the 36 actions, 120 time series were recorded in different environments. In 20 out of the 120 cases, only a target object was placed in the stage, and in the remaining 100 cases, a dummy object was placed in addition to the target object. The dummy object was chosen from 5 objects, and therefore 20 time series were generated for each. The dummy object was never the same as the target, Both a target and an optional dummy object were arranged at random positions within the area range shown by the dashed square in Fig. 2(b). Any arrangement where the target was occluded by the dummy at the home position of the agent was omitted.

Each exemplar time series consists of pairs of visual information and the corresponding motor value computed by a manually coded teaching program. This approach may seem inappropriate. However, if the agent learns the exemplars by rote, there would be no need to learn the action by using the network. The actual objective of the learning is, therefore, to establish the relationships among the provided exemplars in an unsupervised manner. Also, it should be mentioned again that a PB vector has no exemplar.

The teaching program calculates the desired rotation speed of the two wheels of the agent from the position of the specified target taken from a camera image at a constant time interval. At the same time, 27-dimensional visual and 10-dimensional motor information are recorded for later learning (see Fig. 3). The visual input vector does not have the position of the target explicitly. From the viewpoint of the network, the visual field is composed of nine vertically divided regions. Each region is represented by the fraction of the region covered by colored patches and the dominant hue of the patches in the region. The hue is encoded by the position $(\cos\theta, \sin\theta)$ in the color circle, where pure red, yellow, green, and blue are represented as $\theta = 0°, 90°, 180°,$ and $270°$, respectively. The desired speed of the wheel takes a real value ranging from -0.2 to 1.0. A negative value indicates reverse rotation. The motor vector is composed of two five-dimensional real-valued vectors, each of which represents the speed of the wheel in the form of $[f(0), f(0.25), f(0.5), f(0.75), f(1.0)]$, where f is a Gaussian distribution with the mean of the desired speed and a sigma of 0.25. This increases the robustness against the re-generation error of the network.

2.2 Phase 2: Batch Learning

The network learns incomplete parts of the 36 actions in a batch manner by employing the data prepared in the previous phase as the supervising signal. The learning process is formulated as a conventional iterative, steepest descent optimization with respect to the error function E, defined in (1). The model has two types of parameters to be optimized: one is the vector W consisting of all the connection weight values of the network; and the other is the set PB consisting of PB vectors pb_i for all supervised actions $i \in \mathcal{A}$.

$$E(W, PB) = \sum_{i \in \mathcal{A}} E_i(W, pb_i) \tag{1}$$

$$E_i(W, pb_i) = \sum_{j=0}^{119} \sum_{t=0}^{l_{ij}} E_{ij}(t; W, pb_i) \tag{2}$$

$$E_{ij}(t; W, pb_i) = \|\hat{m}_{ij}(t) - m(v_{ij}(t); W, pb_i)\|^2 , \tag{3}$$

where l_{ij} is the length of the j-th training data of an action i, $\hat{m}_{ij}(t)$ is the desired motor vector corresponding to the visual vector $v_{ij}(t)$ at the time step t in the training data, and $m(v_{ij}(t); W, pb_i)$ is its actual value generated by the network under the condition that the connection weight is W, and the PB vector for the action is pb_i with the identical vision input. The parameters W and PB are updated simultaneously by learning all the provided data in a batch manner. The learning procedure is implemented by using the conventional back-propagation algorithm. At the beginning, all the connection weight values are randomized with a small value, and $pb_i, \forall i \in \mathcal{A}$ are set to the zero vector. All the PB vectors reside in the storage since the values of the PB nodes are switched so that the network can learn all the given actions at the same time. And then, the following procedure is conducted 30,000 times.

(1) Do the following for each actions i in $\mathcal{A}$:
  (1.1) Load the stored pb_i to the PB nodes.
  (1.2) For each of the 120 sensor-motor time series, calculate the delta errors of connection weights $\partial E_{ij}/\partial W\,(t; W(T), pb_i(T))$ and of PB vector $\partial E_{ij}/\partial pb_i\,(t; W(T), pb_i(T))$ by using the back-propagation algorithm.
  (1.3) Update pb_i by using the summation of all the delta errors of pb_i for all time steps t of all time-series j of the action i, and store the updated vector to the storage.
(2) Update W by using the summation of all the delta errors of W for all time steps t of all time-series j of all the provided actions i.

Thus, the connection weights capture the common characteristics among all the actions and play a background part while each PB vector is specialized to its corresponding vector. In the analysis of the experimental results, we observe the acquired geometric structure constructed by the PB vectors in the conceptual space.

2.3 Phase 3: Examining the Generalization Capability

Two aspects of the generalization capability of the agent, 1) transfer of the skill to a novel environment and 2) recombination of the supervised actions into an unexperienced action, were tested. For examining the transfer of skill to a novel environment, the agent was tested to determine if it could accomplish each known action in 280 novel environments where a target and dummy object were placed in a systematic manner. The PB vectors acquired through the second phase were employed. This test reveals the kind of information kept in the vectors. If the vector codes only specific trajectories of

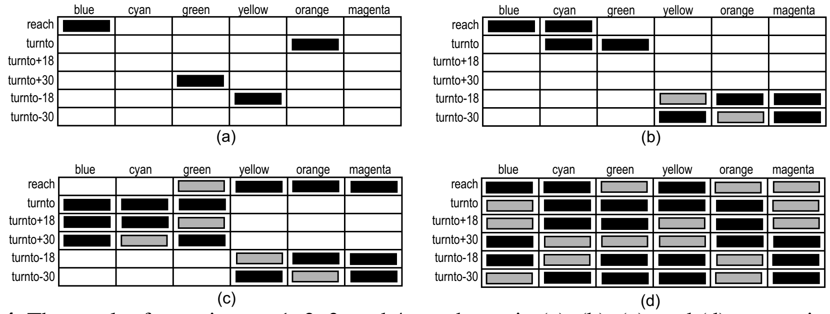


Fig. 4. The result of experiments 1, 2, 3, and 4 are shown in (a), (b), (c), and (d), respectively. A black box represents a trained action, and a gray box represents an action acquired as a recombination of the provided data.

taught examples without generalization, it is impossible to generate a goal-directed action in a different environment. In order to investigate the recombination of supervised actions into an unexperienced action, the PB vector encoding a novel action $i' \notin \mathcal{A}$ should be examined. The vector can be computed by the *recognition* procedure. The algorithm is basically identical to the learning procedure except that W is not updated. By employing 30 out of 120 examples of the action i' produced in the first phase, $pb_{i'}$ is optimized with regard to the error function for the action i' defined in (2) of Section 2.2 by using W acquired in the second phase. Once $pb_{i'}$ is obtained for each unseen action, the generation test can be conducted in the same way as in the trained action cases.

3 Results

We next observe the changes of the generalization capability depending on the sparseness of the provided examples. The degrees of generalization are compared for four sessions of teaching data of different sparseness. In Fig. 4, a trained action is indicated by a black box. In all the experiments, all the trained actions were regenerated successfully; this means the agent could accomplish the goal in more than 80 percent of the test environments explained above. A gray box shows an action achieved by the combinatorial generalization without extra teaching. The criteria of success for the novel action are identical to that for the trained action. In the remaining sections, the results are discussed only from the viewpoint of the performance. We'll re-examine issues about the underlying mechanism in the next section.

Experiment 1: Learning by Rote In this case, no combinatorial generalization was observed because of very sparse training data (Fig. 4(a)). This suggests that the agent regarded the provided actions as being holistic; namely, it could not find any re-usable parts such as an operation and a target.

Experiment 2: A Local Compositional System As training data increases, two novel actions `turnto-yellow-18` and `turnto-orange-30` were acquired without learning exemplars (Fig. 4(b)). This implies that the local compositional system is self-organized since one of the reusable operations `turnto-18` and `turnto-30` and one of the reusable targets `yellow`, `orange`, and `magenta` could be composed in any possible way, including unseen ones. However, the agent could not acquire `reach-green` and `turnto-blue`.

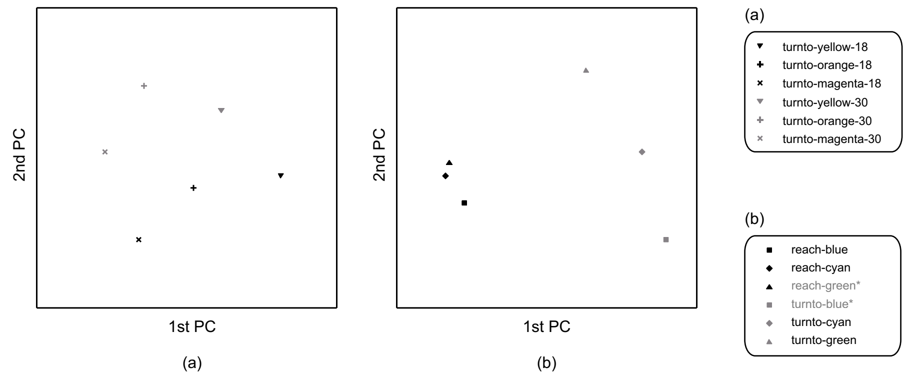


Fig. 5. Two parts of the concept space self-organized in experiment 2 are presented. Actions depicted in (a) constitute a geometric regularity as the underlying structure of the compositional system, whereas the actions depicted in (b) do not, as the agent failed to generate actions marked by *. Note: PC means principal component.

Experiment 3: Two Independent Compositional Systems / Categorization Two separate local compositional systems emerge when further training data were added. One is system { `turnto`, `turnto+18`, `turnto+30` } × { `blue`, `cyan`, `green` }, and the other is system { `reach`, `turnto-18`, `turnto-30` } × { `yellow`, `orange`, `magenta` }. They are independent of each other since the targets of one system cannot be applied to the operations of the other system. The result can be interpreted as the categorization of targets based on operations applicable to these targets (Takamuku, Takahashi, & Asada, 2006). It should be noted that it is impossible for a model employing predefined roles of elemental concepts to realize this type of generalization. If two classes corresponding to a target and an operation are defined *a priori*, only one global compositional system could emerge.

Experiment 4: Operation (Target, Offset) Finally, all the possible actions were acquired when the robot was trained with examples consisting of 21 out of the 36 actions. At least two incompatible interpretations of the results are possible. One is that all six operations have an equal relationship with one another. In this case, each operation is regarded as being a discrete symbol. The other possibility is that the similarity based on the offset values is understood. If so, the operations concerning `turnto` could have the structure `turnto` × { `-30`, `-18`, `0`, `+18`, `+30` }, and the `reach` operation exists apart of them. The operations should be divided into two classes `TURNTO` and `REACH` in terms of the conceptual structure. The result of an additional experiment proved that the latter interpretation is correct: the similarity based on the offset values was understood. The agent could re-generate some actions which have intermediate offsets such as `turnto-blue-24` by recognizing newly created examples of the actions. Last but not least, all six targets form a `TARGET` class, since they can be applied to all operations equally. Thus, the roles of `TARGET`, `OPERATION` (= `TURNTO` + `REACH`), and `OFFSET` emerge to organize the argument structure `TARGET` × ((`TURNTO` × `OFFSET`) + `REACH`).

4 Analysis

For the analysis of the acquired structure in the PB space, we discuss the underlying mechanism of the combinatorial generalization proposed in Section 1: the geometric

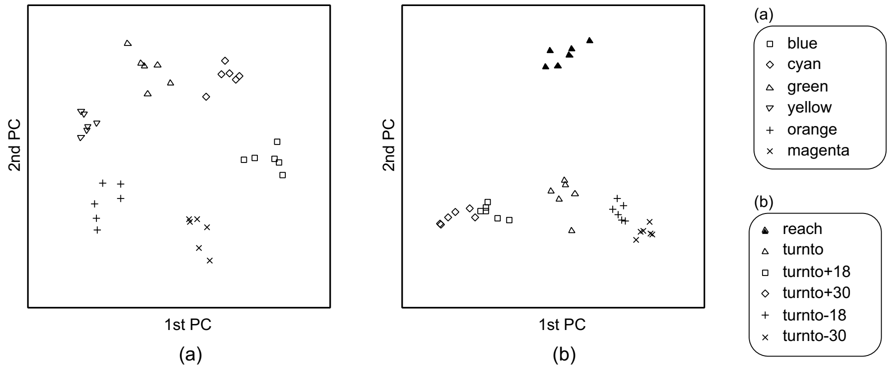


Fig. 6. The concept space self-organized in experiment 4. The space is projected to the planes in which the difference of 36 actions with respect to targets (a) and operations (b) are maximized.

regularity self-organized in the conceptual space. A main objective of the discussion is to bridge the gap between the symbolic behavior of the system and its sub-symbolic implementation.

Figure 5(a) shows a concept structure underlying the local compositional system observed in the second experiment. PB vectors for six actions included in the system are displayed (see also Fig. 4(b)). The displayed vectors are obtained through the learning process for a trained action and through the recognition process for an untrained one. The original 12-dimensional vectors are projected onto a 2-dimensional plane computed by applying the conventional principal component analysis (PCA) method to the six vectors. The accumulated contribution rate up to the second principal component (PC) is 0.79. A regular structure similar to the prism shown in Fig. 1 is observed in the figure, although the third and subsequent PCs show irregularity. Thus, an unexperienced action has the "correct" position in the concept space. Two congruent triangles representing the two elements `turnto-18` and `turnto-30` are regarded as being equivalent in terms of their relation to the three elements `yellow`, `orange`, and `magenta`. This is a possible underlying representation of role-governed categories (Markman & Stilwell, 2001) of elements: these two groups have the role of an operation and a target, respectively. It should be noted that both roles emerge at the same time since both are defined in a circular manner. This is well represented geometrically in that the congruency of two triangles always accompanies the congruency of three lines corresponding to each of the targets. Meanwhile, no regularity is found in the plot of actions { `reach`, `turnto` } × { `blue`, `cyan`, `green` } (Fig. 5(b)). This is consistent with the performance that no combinatorial generalization was realized with regard to the abovementioned actions. A similar picture is found in the first experiment, where no generalization was realized.

In the third experiment, we can find two separate regular structures in accordance with the observed performance. They exist on different sub-spaces, although these sub-spaces are not orthogonal to each other. This explains the incompatibility of elements between the systems. Each of the systems utilizes its dedicated representation of an element and a composition mechanism.

A new facet is discovered in the concept structure self-organized in the last experiment. Not only a structure representing the relationships among elements of different roles but also one representing the similarity among elements within each role are observed clearly. The former is the congruency of sub-units, which is similar to the structure found in the second and third experiments. The latter, with regard to each role of a target and an operation, are shown in Figs. 6(a) and (b), respectively.

The projection plane of Fig. 6(a) is chosen by applying PCA to representative vectors of the targets obtained by averaging the PB vectors for all operations for each of the targets. If a component of a target and an operation in the PB vectors are independent of each other, this method averages away the operation information. This assumption is shown to be true later in this paper. The accumulated contribution rate up to the second PC is more than 0.98, and so almost all the information is displayed in the plot. Six clusters corresponding to each of the targets are observed in the figure. This implies that each target has its own representation in the subspace regardless of its surrounding context; namely, an operation takes the target as its argument. This can also be stated as follows: a subspace holding information of a specific role, a target in this case, emerges. Furthermore, the clusters are arranged in a circle like the continuum of color by hue. This arrangement suggests that the agent understands the similarity of color of the target. This is indirectly proven by the tendency to choose a target of a closely related but incorrect color. The more similar a dummy object is to a target with respect to color, the more easily the agent mistakes the dummy for the target. And so, the generalization of color is realized. When there is no specified target in the environment, the agent chooses an object that has a color similar to the target as a substitute.

In Fig. 6(b), the projection plane to see the difference among the operations is chosen by averaging the target information instead of the operation information. The accumulated contribution rate up to the second PC is more than 0.86. Here, both continuous and discrete sub-structures exist at the same time. In the first PC (x-axis of the figure), the continuum of operations `turnto`s by offset emerges. Apart from that, the cluster of the operation `reach` is positioned. This implies that the second PC (y-axis of the figure) carries the distinction between `reach` and `turnto`. In addition, another continuum of the `turnto` operations by the absolute value of the offsets is found in the third PC, of which the contribution rate is approximately 0.12. Thus, the subspace of operations consists of three orthogonal components. In addition, the subspace of both targets and operations are also orthogonal to each other, since the cosine between any pair of vectors taken from both subspaces is less than 0.01.

5 Discussion and Conclusion

We discover at last an underlying analog mechanism of the phenomenologic system of symbols inferred in Section 3 by considering the following correspondences:

(1) The analog correspondence of an elemental symbol is the center of gravity of a cluster of actions containing the element as a part (see Figs. 6(a) and (b)).

(2) The composition of symbols is realized by summing up their corresponding vectors.

It should be remembered that the conceptual elements are *dependent* on each other, although the vectors representing them are *independent* of each other. The superficial independence of elements strongly relies on the structure-preserving map between a PB vector and an action, which self-organizes in the connections of the network. The map provides all the fundamental devices to maintain the functional compositionality of the system, such as the composition rule and the role-governed categorization of elements.

Without them, the conceptual elements cannot constitute an embodied whole action, just as chess pieces without a chessboard cannot constitute a game.

The above discussion can be transferred to a conventional symbolic system by replacing the independence of vectors with the atomicity of symbols. We usually think that a symbol carries its meaning independently; however, the discussion suggests that this idea is based on a lack of attention to the existence of the background mechanism. Keeping the connection between a symbol and its referent in the real world is not enough to maintain the coherency between an internal compositional system and the reality outside. The symbol grounding problem proposed by Harnad (1990) should be reconsidered from the broader perspective of a system in which both symbols and their background are essentially inter-related.

To conclude, a sub-symbolic implementation of the recombination of goal-directed actions is presented. Three different types of functionally compositional systems emerge depending on the sparseness of the provided examples by using an identical learning model. The predefined features are 1) the architecture of the connectionist model and 2) the categorization of the examples, namely, with which PB vectors the network should learn each example. The information on the categorization does not provide any relationships among the categories. In future work, the transition process of the internal structure involved in the transition from a holistic system to a compositional one will be presented. Also, associative learning between goal-directed actions and sentences will be investigated by employing the technique proposed in Sugita and Tani (2005).

References

Cangelosi, A., & Riga, T. (2006). An embodied model for sensorimotor grounding and grounding transfer: Experiments with epigenetic robots. *Cognitive Science*, *30*(4), 673-689.

Goldberg, A. (1995). *Constructions: A Construction Grammar Approach to Argument Structure.* University of Chicago Press.

Harnad, S. (1990). The symbol grounding problem. *Physica D*, *42*, 335–346.

Iwahashi, N. (2006). Robots That Learn Language: Developmental Approach to Human-Machine Conversations. In P. Vogt, Y. Sugita, E. Tuci, & C. Nehaniv (Eds.), *Symbol Grounding and Beyond* (pp. 143–167). Springer-Verlag.

Markman, A., & Stilwell, C. (2001). Role-governed categories. *Journal of Experimental and Theoretical Artificial Intelligence*, *13*(4), 329–358.

Pollack, J. (1991). The induction of dynamical recognizers. *Machine Learning*, *7*, 227–252.

Roy, D. (2002). Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language*, *16*, 353–385.

Skinner, B. (1957). *VERBAL BEHAVIOR.* B.F.Skinner Foundation.

Sugita, Y., & Tani, J. (2005). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior*, *13*(1), 33–52.

Takamuku, S., Takahashi, Y., & Asada, M. (2006). Lexicon acquisition based on object-oriented behavior learning. *Advanced Robotics*, *20*(10), 1127–1145.

Tani, J., Ito, M., & Sugita, Y. (2004). Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB. *Neural Networks*, *17*, 1273–1289.

Tomasello, M. (2003). *Constructing a Language: A Usage-Based Theory of Language Acquisition.* Harvard University Press.

van Gelder, T. (1990). Compositionality: A connectionist variation on a classical theme. *Cognitive Science*, *14*, 335-364.