DARWIN 270138

Dextrous Assembler Robot
Working with Embodied Intelligence

SEVENTH FRAMEWORK PROGRAMME
ICT Priority

Deliverable D4.3: “Growing concept system for robotic agents”

Due date of deliverable: 31.01.2013
Submission date: 15.02.2013
Re-submission date: 17.02.2015

Start date of project: 1st February 2011
Duration: 48 months

Coordinator:
Dr. Christian Eitzinger
Profactor GmbH

Lead Beneficiary: NOCA (S. Kasderidis)
Contributions by: IIT (V. Mohan)

---

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th>Project co-funded by the European Commission within the 7th Framework Programme</th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>X</td>
</tr>
<tr>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
</tr>
<tr>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission Services)</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
</tbody>
</table>
# Contents

**PREAMBLE** ................................................................................................................................. 4

**1.0 INTRODUCTION** .................................................................................................................. 7

**2.0 BACKGROUND ON CONCEPTS** ......................................................................................... 11

2.1 PHILOSOPHY OF CONCEPTS ... ................................................................................................. 11

2.2 PSYCHOLOGY OF CONCEPTS ...................................................................................................... 12

2.2.1 Main Properties ........................................................................................................................ 12

2.3 MAIN THEORIES ....................................................................................................................... 13

2.3.1 Review of Exemplar theory ..................................................................................................... 13

2.3.2 Review of Prototype Theory ................................................................................................... 14

2.3.3 Typicality and Context Effects ............................................................................................... 14

2.3.4 Review of Theory-Theory ....................................................................................................... 16

2.4 ASSIMILATION LEARNING THEORY AND “CONCEPT MAPS” ......................................................... 16

**3.0 A MODEL OF SPATIAL RELATIONS** .................................................................................... 19

3.1 PAIR WISE SPATIAL RELATIONS IN DARWIN ........................................................................... 20

3.1.1 Orientation Comparison .......................................................................................................... 23

3.1.2 Topological Features ................................................................................................................ 26

3.1.3 Learning of spatial relations ................................................................................................... 28

3.2 SIMULATION RESULTS ............................................................................................................ 28

**4.0 A REPRESENTATIONAL FRAMEWORK FOR CONCEPTS** .................................................. 32

4.1 A COMPUTATIONAL CONCEPT REPRESENTATION FRAMEWORK ............................................ 32

4.1.1 Key Ideas ................................................................................................................................. 32

4.1.2 Overview of the global concept model ....................................................................................... 36

4.2 REPRESENTATIONAL LEVELS, CLASSIFICATION AND STORAGE-RECALL OF SEMANTIC KNOWLEDGE ....................................................................................................................... 41

4.2.1 Exemplar Representation ......................................................................................................... 41

4.2.2 Prototype representation ........................................................................................................ 44

Adding a new Concept ..................................................................................................................... 45

Incremental Change of a Concept .................................................................................................... 48

Change of Summary Feature Values ............................................................................................... 48

Change of Exemplars ....................................................................................................................... 49

Controlling the Capacity of the Supporting Sets ............................................................................ 49
4.2.3 Capturing semantic knowledge ................................................................. 50
4.3 SIMULATION RESULTS ................................................................................. 53
  4.3.1 Visual System input and DARWIN objects ........................................... 53
  4.3.2 Experiments in DARWIN ...................................................................... 57

5.0 BRAIN-GUIDED ORGANIZATION OF CONCEPTS IN DARWIN ............ 74

  5.1 DISTRIBUTED "PROPERTY SPECIFIC" ORGANIZATION OF OBJECT CONCEPTS: LEARNING, REASONING AND GROWING AS IT TAKES PLACE ................................................................. 75
    5.1.1 Learning and self organization of color, words and shape related information .................... 76
    5.1.2 Network dynamics, Pattern completion and Multimodality .................................................... 83
    5.1.3 Bottom Up vs. Top Down neural activations and resulting inferential capabilities .................. 85
    5.1.4 Detecting Contradictions: Switching to attention driven exploration to learn further ............ 87

  5.2 CONNECTING OBJECTS, ACTIONS AND THE BODY SCHEMA ....................... 89

6.0 CONCLUSIONS ......................................................................................... 94

REFERENCES .................................................................................................... 97
Preamble

This deliverable in its revised form has been prepared after taking into account the comments of the DARWIN Y3 review. According to these the assessment of the deliverable is:

Y3 Review comments
D4.3 was rejected in Y2, submitted “revised”, but the report seems to be the same, with some restructuring, re-arrangements and formatting changes, and an added section on spatial relations. The corrections requested by the reviewers of Y2 were not implemented.

Y2 Review comments
Report suffers from a lack of coherent architectural ideas. For instance, it remains unclear how the object perception architecture can support certain functionalities required (pose estimation) with the necessary quality of service attributes: precision, scalability, etc. Integration of the ideas described here with work by CVUT and FORTH is not clear.

To aid the reviewers in their work we present briefly the contributions and changes that have been made to the deliverable in order to easily track and check the content of the new version.

Answers to the main Y2-Y3 observations follow below.

Answers to Y2-Y3 Comments
Regarding the comment about the lack of coherence of the architectural ideas we believe that the difficulty might come from two sources: One is the impression that the concept system is also related to the visual recognition of objects and the other is the relative complexity of its internal dynamics. For the first point we have to clarify that the concept system is not involved directly in object detection and identification. This is the task of the perceptual system. The latter is responsible for acquiring various appearance-based properties of objects such as size, pose, colour, texture (if applicable) and shape class. In principle the perceptual system could also provide additional information channels such as object affordances and actions taken on them, world state, etc. All the above information channels and other internal ones (such as set goals, internal robotic body context, state of reasoning system, etc) provide the overall input to the concept system in the form of sets of property-value pairs which constitute the system percepts of the internal and external world. The main function of the concept system is twofold. First is to discover percept regularities (in an unsupervised way), to classify them under internal class labels and to store subsequently any incremental knowledge regarding the corresponding classes (concepts) that might arise. Second it must return the stored knowledge about the
concepts in question to the reasoning system or other user (human or UI enquiries) filtered for the right context. In principle the concept system could help the perceptual system in object appearance-based properties determination (e.g. shape class) by generating and delivering class-label hypotheses to the latter. The link to achieve this coupling was through the visual attention system. However as this work was delayed, and not realised until end of Y3, there was not a feedback route between the concept and perception systems; only the direct one.

Regarding the issue of the accuracy of the appearance-based properties in forming and delivering useful concepts the answer is rather related to the requirements of robot manipulation accuracy vs. robot perception capabilities. The concept system only serves here as placeholder of such information and not as its final user as it is case for the motor system. Thus it is not actually impacted by the accuracy requirements. Its main usefulness is to organise the received information in a way that is most discriminant and at the same time compact regarding a given problem domain and the related context. If this problem is object manipulation and assembly or the classification of texts in a corpus of literature is a secondary matter regarding the main design goals of the concept system as, by its nature, it should be domain agnostic and based on a set of ideas that allow wide application on many specific problem domains.

On the issue of the architectural complexity we tried to enhance substantially the material of the report in order to explain a non-trivial information processing dynamics of the concept system. For this reason new sections were added, more simulation results are presented and better explanations are provided as to the information interface of the perceptual and concept systems. The latter clarifies also the integration with the work of the CVUT and FORTH partners. For detailed information please see the changes table below.

**Document Changes Table**

<table>
<thead>
<tr>
<th>S/N</th>
<th>Section</th>
<th>Change</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>Updated</td>
<td>Added reference to Ausubel’s Assimilation Learning Theory and the integrated Theory-Theory view</td>
</tr>
<tr>
<td>2</td>
<td>2.2</td>
<td>Updated</td>
<td>Section substantially decreased to include main properties only</td>
</tr>
<tr>
<td>3</td>
<td>2.4</td>
<td>Changed</td>
<td>Section now includes the basis of the Theory-theory view included in the model</td>
</tr>
<tr>
<td>Section</td>
<td>Subsection</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>3.2</td>
<td>Updated</td>
<td>Section includes clarifications of simulation results and a similarity function for comparing composite object descriptors</td>
</tr>
<tr>
<td>5</td>
<td>4.1.2</td>
<td>Changed</td>
<td>Section now includes an overview of the information processing of the integrated model to aid comprehension</td>
</tr>
<tr>
<td>6</td>
<td>4.2</td>
<td>Changed</td>
<td>Section now includes the sub-models for exemplar and prototype spaces (4.2.1-4.2.2). Section 4.2.3 describes the theory-theory model.</td>
</tr>
<tr>
<td>7</td>
<td>4.3</td>
<td>Changed</td>
<td>Substantially re-written to include section on interface of perceptual-concept system (section 4.3.1) and extended section on experiments (4.3.2)</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>Updated</td>
<td>Section is updated regarding new addition of the theory-theory view and open issues</td>
</tr>
<tr>
<td>9</td>
<td>References</td>
<td>Updated</td>
<td>More references were added, mostly on theory-theory view and causal networks (37-42)</td>
</tr>
</tbody>
</table>
1.0 Introduction

This deliverable presents the current progress in the development of the DARWIN concept system. It covers work of the second year and developments of the third year of DARWIN. In doing so, it takes a more general view in the problem of representation considering representation issues that are slightly wider than the needs of DARWIN. The motivation behind this is that representation and discovery, represented by the concept and reasoning systems respectively, are two of the centrepieces of any architecture for cognition.

One has to ask the question why concepts are important in the context of a cognitive architecture and in DARWIN in particular. The answer to this question is the fact that concepts give to a cognitive agent the ability to consider not only perceptual information but also previous perceptual and non-perceptual experiences and knowledge possibly suppressing current perceptual information when needed (e.g. when perception contradicts beliefs encoded through the conceptual representations) [1]. This provides a flexibility that goes beyond simple reactive or direct perception-action architectures. In addition the contradiction in expectations drives the agent to learn more about its environment in order to minimise the discrepancies. The regular structure of the environment is typically organised through Naïve Domain Theories. These in turn are based on conceptual representations that capture the causal structure and observed regularities of the external world. Perceptual representations cannot capture causality; in fact perceptual interpretation is an ill-posed problem and perceptual “recognition” needs to make additional assumptions about causality and regularity in the external world in order to handle the interpretation problem.

It should be made clear at this point that the deliverable discusses mainly the representational problem and not that of discovery. The distinction is important
because causal relation discovery can only take place in the reasoning system after observing the results of the agent actions in the environment. In fact in deliverable D4.2 (Initial Reasoning System) there is a nice presentation of a mechanism for (apparent) causality discovery; for example which perceptual properties, such as size, shape or colour, influence the motion of an object. The comparison of observations with expectations leads to the decrease of importance of the colour property until it becomes irrelevant with increasing experience. This fact is then represented in the concept system.

Some general important questions in devising a representational scheme are always the nature of variables involved (e.g. numerical vs. categorical), the representation of similarity relations between entities of a subject domain, the use of local or distributed representations, the self-organisation of the concept system as perceptual input is received throughout the lifetime of the agent, generalisation performance (especially in novelty situations and across domains) and the lifetime of the conceptual representations themselves. All the above issues are strongly related to, and the development should be guided by, the primary usages of a concept system; mainly that of Categorisation, Provision of Default Values (for a class representative or missing values/features for an instance), Inference, Association and Analogical Reasoning.

Concepts encode semantic knowledge, i.e. information related to the causal structure and observed regularities of the external world. Thus representations such as those of state of agent/environment, episodic or procedural memory are not considered as conceptual representations even though they are paramount in the operation of a cognitive system. In addition to the above, conceptual representations can be thought as a kaleidoscope in the sense that depending on the reasoning context, i.e. on the particular view of the information contained in a conceptual representation, the provided information is different; filtered by the particular reasoning need at the time. For example, for material objects, one can request properties related to object
appearance (i.e. perceptual-related information – an “IS / LOOKS” context), to object affordances (the set of potential actions that are valid for the given object/agent combination), to causal relations between properties and actions, etc. While the conceptual representation of an “object” includes different kinds of information, nevertheless the “object” itself is more like a vehicle of properties, which come into focus in the various reasoning contexts, rather than the origin of such properties. This is a subtle point: “objects” are entities with properties that we attribute to them either perceptually (when possible) or mentally by some reasoning and examination process. It is well known, in fact, that one can attribute infinite properties to an “object” [2]. What is challenging for the development of a conceptual representation is to find and encode only those properties which are relevant in a given reasoning context; hopefully they will be a finite set. The mechanism to do so is the extraction of causal and apparent structure building eventually Theories of Domain [2]. Our approach in the representational problem is guided by research in the fields of philosophy and psychology of concepts and neuroscience.

In section 2 we present briefly known facts from the literature, especially from the fields of psychology and philosophy of concepts and educational psychology. We outline there the main properties of conceptual representations as they were observed from numerous psychological experiments. We present briefly Ausubel’s Assimilation Learning Theory which explains conceptual development and provides a model for representation of concepts under the Theory-Theory approach [41]. In section 3 we present a model for extracting spatial relations between two objects of interest. The motivation behind it is to capture part-whole relations (from perceptual information) and thus to be able to represent composite object concepts. Thus it can be considered as a concept-system sub-process. In section 4 we present a representational framework which tackles most of the issues presented in section 2. We formulate the key ideas of the framework, providing an initial concrete representational model but the discussion is general enough so as other concrete representational systems to be derived as
needed. We provide a set of simulation results to assess the representational framework. In this section we show how the representations can be used for classification, storing and recall of semantic knowledge tasks. In section 5 we present the concept system which has been currently implemented in the Y2-Y3 demonstrators. Model presentation and simulation results are given. In section 6 we conclude with some points of interest and open issues.
2.0 Background on Concepts

In this section we will provide a brief overview about important findings from the study of concepts in the fields of philosophy and psychology of concepts. We will also present briefly the Assimilation Learning Theory which leads to the notion of a “cognitive map”. The material aims to summarise the main properties of conceptual representations so as to guide the design of a representational framework in section 4 and the implementation of the DARWIN concept system in section 5.

2.1 Philosophy of Concepts

Concepts are highly researched in philosophy, cognitive science and linguistics [3, 2, 4, 5]. There are a wealth of theories, viewpoints and techniques in which one approaches the problem of conceptual representations. As a starting point, of discussion for the problem of representation, one can start from philosopher Armstrong’s Observations (1978), [3]:

1. The members of the two classes all have something in common (they are all shapes, they are all colours);
2. But while they have something in common, they differ in that very respect (they all differ as shapes, they all differ as colours);
3. They exhibit a resemblance order upon their intrinsic nature (triangularity is like circularity, redness is more like orangeness than redness is like blueness), where closeness of resemblance has a limit in identity;
4. They form a set of incompatibles (the same particular cannot be simultaneously triangular and circular, or red and blue all over).

These are the essential properties that any conceptual representation should respect. Moving further, most researchers nowadays assume that a concept is represented by the use of domains (properties), even though there is some critic in this idea [4]. In addition it is assumed that each domain has a different saliency for a given concept depending on the context. Generalising a concept definition from [4] we can say that:
A concept is represented as a set of regions in a number of domains together with an assignment of salience weights to the domains and information about how the regions in different domains are correlated.

For example in the APPLE case, there is very strong (positive) correlation between the sweetness in the taste domain and the sugar content in the nutrition domain and a weaker correlation between the colour red and a sweet taste.

At this point one has to clarify the difference between properties (or features) and concepts. Properties are single-domain concepts, while concepts are typically multi-domain constructs with saliency weights for domains that depend on the context.

### 2.2 Psychology of Concepts

In this section we present briefly important properties that conceptual representations possess in humans. For more details the interested reader should see [1] where relevant experiments are described. In section 2.3 the main psychological theories of concepts are covered briefly, while in section 2.4 Ausbel’s Assimilation Learning Theory is presented. The former provides the basis for the idea of the “concept map” which in turn is a model for the Theory-Theory representational approach.

#### 2.2.1 Main Properties

According to experimental findings, and also additional observations from the literature [1, 2, 4] we can summarise the main properties for conceptual representations as:

- **P1**: Concepts form incrementally;
- **P2**: There are typicality effects present;
- **P3**: Concepts depend on context, i.e. their essential features change depending on the context of use;
- **P4**: A given instance may belong at the same time to a concept and a more abstract one but also not have a lot of commonalities with other members of the abstract class;
- **P5**: Transitive inference might not always hold;
13

• P6: Concepts combine to produce new concepts with emergent properties that might not be predicted by the knowledge of properties of the constituent concepts.

These properties will be used to guide the design of a representational scheme in section 4.

2.3 Main theories

In this section we briefly present the three main theories which provide a guide on the representation of concepts.

2.3.1 Review of Exemplar theory

According to Exemplar theory concept formation takes place by memorising observed instances. Consequently instance classification takes place by comparing the new instance to the set of stored instances. In practice there are many Exemplar theories, as each one specifies different rules as to which instances are stored, how many are retained, and what is the set of instances that the new input compares against. Typical classification rules for deciding the class of a new instance include:

• The calculation of similarity of instance (i), against all stored instances and then the calculation of ratios for each category. These ratios are used as choice probabilities for the final selection. For example a new instance might be a CAT with a probability equal to (assuming only two classes CAT and DOG):

$$P_i(CAT) = \frac{\sum s_{ij}(j=CAT)}{\sum s_{ij}(j=CAT)+\sum s_{ij}(j=DOG)} \quad (1)$$

Where $s_{ij}$ represents the similarity between instances i and j;

• The ALCOVE model of Kruskhe, [6], uses a weighted sum of all similarities. It has formulated as a neural network model for parallel calculation of similarities with back-error propagation for learning of class probabilities from similarities;
• K-neighbours classification (K-NN), where the class selected is the one that dominates
  the class labels of the K-closest neighbours;
• Nearest Neighbour (1-NN) classification, where the class assigned is the class of the
  closest neighbour. This rule has good support in many experimental settings; see [2].

Deficiencies of the exemplar theory include the lack of prototypes and thus difficulty in
constructing abstractions and concept combinations. It has been found experimentally that
there are people that use an exemplar-based strategy in classification tasks.

2.3.2 Review of Prototype Theory
Prototype theory suggests the existence of summary representations, which encode the general
information about the class, which is not necessary present in any given instance. The theory
assumes that a schema structure is often in place. Schema is a data structure with a number of
slots (called frames); each one of them corresponds to a feature of the schema. The feature
values of the instances are stored in the corresponding frame of the schema. Usually
information regarding the probability distribution out of which the values are drawn is stored as
well in each frame.

This type of theory is strong in explaining abstractions and at least some cases of concept
combination effects. However, the schema approach cannot easily incorporate constrains and
correlations among the frame values, nor it can easily provide a set of prototypes when multiple
values are common for each schema feature. See [2], chapter 2, for further information.

2.3.3 Typicality and Context Effects
It has been observed experimentally that there are instances that are somehow more typical
than others as representatives of a class, while others seem to be almost border cases between
two classes. An example is the cases of chair and telephone as furniture items at home. It has
also been observed that contextual effects exist. For example one typically thinks of cows, in a
To account for these phenomena psychologists have postulated the existence of suitable metric spaces (which are called conceptual spaces, [4]) where the notion of a distance function between two instances can be represented. Assuming for the moment that the feature vectors consist only of numerical variables, one can easily use a Euclidean distance function as a possible distance measure:

\[
D(i, j) = \sqrt{\sum_k (x_{ik} - x_{jk})^2}
\] (2)

Where, \(x_{ik}\) and \(x_{jk}\) are the components of the features vectors \(x_i\) and \(x_j\) corresponding respectively to instances \(i\) and \(j\). Contextual effects can be explained by assuming a suitable weighting for the component domains, i.e. for the Euclidean metric:

\[
D(i, j) = \sqrt{\sum_k w_k (x_{ik} - x_{jk})^2}
\] (3)

where the coefficients \(w_k\) are called attention coefficients (in the psychological literature) and they encode the relative importance of the various components. Different sets of the attention coefficients represent different contexts. Thus in the example of horses and cows and assuming that the sets of their attributes also include attributes for riding and milking it should be evident that by changing the values of the corresponding attention coefficients for the riding and milking attributes, when in a food or riding context, we can introduce different distances of two animal instances from a prototype animal which has also the attributes of milking and riding.

This naturally leads us to consider the notion of similarity between any two instances. It has been proposed that similarity is a function of distance of any two instances. Usually it is assumed to be a Gaussian or (negative) exponential function as in (4) and (5):

\[
s_{ij} = \exp(-c \cdot D(i, j)^2)
\] (4)

or
\[ s_{ij} = \exp(-c \ast D(i,j)) \] (5)

where \( c \) is a variable called specificity and controls the effective size of the neighbourhood that any two instances are considered similar enough.

It is now easy to explain how typicality effects arise. Assuming that somehow we have some instances (or prototypes) that are considered representative of the class, then any other given instance can be judged to be similar enough or not. As closer as an instance is to a prototype the more easily it will be recalled. For example, when one recalls birds usually consider first (as a typical prototype) a robin rather than a penguin.

### 2.3.4 Review of Theory-Theory

The basic tenet of the theory-theory is that semantic cognition is constrained to a large extend by the naive domain knowledge – often referred to as a “theory” – that people hold about the causal relations that exist between entities in a domain. A key function of a naive domain theory is to explain and predict observed events in the environment. Domain theories serve this function with reference to stored knowledge about the causal properties of (and causal relations between) objects. According to this view children and adults make sense of the sea of information that constitutes the environment by employing domain-specific constraints that specify the causal relations between objects and their essential properties. Most researchers in this tradition seem to believe that theory-theory is a complimentary approach that can help categorisation-based theories overcome some of their limitations. From this point of view, categorisation-based theories, of sections 2.3.1 and 2.3.2, in and of themselves are thought to provide an inadequate framework for explaining semantic cognition without further constraints to determine which categories and properties are stored in memory, and how these are consulted in particular tasks.

### 2.4 Assimilation Learning Theory and “Concept Maps”

Ausubel was an educational psychologist studying the way young children develop their knowledge structures from infancy to adulthood. In contrast to theories from Piaget and Vigotsky he believed that young children do not differ significantly from adults in their cognitive
capabilities in acquiring knowledge, thus he didn’t accept the four developmental stages of Piaget. He attributed the differences between children and adults in the difference in experience between the two types of learners. An adult has much more experience and learning chances than a child thus he develops richer knowledge structures than a child which in turn help him to interpret the world around him in a more predictable and efficient way. There are various sub-processes that take place during “meaningful learning” such as subsumption, forgetting and progressive differentiation. A basic posit of his theory is that concept development can take either of two forms:

- **Concept Formation**: By 30 months most children have recognized and accurately learned to label 200-300 regularities or patterns. The use of language labels to designate regularity is called concept formation. Typically primary concepts are acquired (i.e. mainly with perceptual means).

- **Concept Assimilation**: The child acquires secondary concepts through the process of concept assimilation. Here concepts and propositions in the child’s cognitive structure are used to acquire new concept meanings, including concepts that have no visible exemplars such as molecule, love and history. New concepts are acquired by using spoken or written words and propositions that already have meaning for the learner. By school age almost all concept learning is concept assimilation.

His theory has found strong experimental evidence during 1960-1970s.

His ideas were taken and extended further by J. Novak which introduced the “Concepts Map” model [41]. According to this human knowledge is composed of four elements: facts, concepts, propositions and principles. Facts are events we observe in the environment or regularities of objects. When two or more concepts are related by the use of “linking words” propositions are formed. Principles are relations between concepts. Principles tell us how events or objects work or how they are structured. Concepts are regularities or patterns in events or objects or records of events or objects, designated by a symbol. Putting all these elements together we can construct a visual representation of knowledge which Novak calls a “Concept Map”. In figure 2.1 we see a concept map drawn by a six-year old student (‘Denny’) (figure 5.4 from [41]).
Fig 2.1: Concept map as drawn by a six-year old student (‘Denny’) given the words {water, solid, liquid, gas, vapour, river, ice, steam}. Observe that the structure of the map is actually a set of propositions describing probabilistic relations between concepts.

Observe that the map is actually a graph structure with concepts as nodes and relational terms as edges. This structure is very reminiscent of the Bayesian graphical causal models [37, 38, 39, 40] which is an abstraction of this model. What is the important message of this section is that the “concept map” model could be used as a representational strategy under the theory-theory approach, thus allowing us to develop a unified representational framework in section 4.0 where all three approaches of representation are covered in the DARWIN concept system.
3.0 A model of spatial relations

In this section we will describe a model of pair wise spatial relations. The motivation behind the development of such a model is the assumption that a complex object, that consists of a number of components, can be described as a tree of components where each link connects a component with another component using a particular spatial relation. Thus the basis of the composite object representation lays to the development of a model for pair wise spatial relations between object parts. An additional assumption is that the component parts can be recognised, by the visual system, on an individual basis as standalone pieces. In the case where a part is already embedded in an assembly it is assumed that the part can still be recognised when the partial occlusion is within reasonable values. If this is not the case, it can be assumed that using a different viewing angle will allow the DARWIN system to identify a component successfully in contrast to failed recognition from a previous angle with high occlusion of the component in question. In case a component is already fully occluded, because it can be perhaps in the internals of a complex assembly and it is out of sight currently, it is assumed that the component is already identified and its representation has already entered in the composite object representation in a previous step. Thus in every case it is possible to construct a composite object representation starting from the individual pieces and progressing step-by-step to the assembly of the target object.

While the model of pair wise spatial relations is not related directly to the concept system it is still a processing stage before the concept system where one extracts the complete tree of whole-parts structural relations. This in turn is nothing more than a description for the (complex) „shape“ feature for a composite object. The assumption is that the individual components can be recognised atomically from various points of view while the composite object can be described by the tree of spatial relations. Our approach in describing the „shape“ of a complex object is based only on sensory information and thus is in principle applicable to any object, novel or already known. An alternative representation is based on motor information where the object is coded as a sequence of commands needed in order to produce the composite object through basic motor assembly commands. The two representations are complimentary and serve different purposes. The former works for any object irrelevant of the fact if we have already assembled it or not. In addition whole structural parts can be linked to
semantic information that might be of particular importance for the components in question. The latter exists after a successful assembly of the object in question. It is more suitable to effectively guide the assembly of known objects.

In sub-section 3.1 we present the core model of the spatial relations. In section 3.2 we provide simulation results from the objects used in DARWIN during year two of the project. The model development was based on ideas presented in [1, 35, 36] and references therein.

3.1 Pair wise spatial relations in DARWIN

The model of spatial relations is a type of back-propagation network which is trained to learn a number of relations between two objects. The first object is called the Reference Object (RO) and the other is called the Secondary Object (SO). The roles of the objects are important due to the interpretation of the spatial relation. This follows the general description:

\[ <SO> \text{ is } <RELATED> \text{ to } <RO> \]  \hspace{1cm} (6)

where the placeholders \(<SO>\) and \(<RO>\) serve as the identifiers (or in general the representations) for the secondary and reference objects respectively. The \(<RELATED>\) placeholder refers to the type of spatial relation that holds between the objects. The relations that are currently supported by the model are the ones appearing in the following list and their combinations:

1. LEFT
2. RIGHT
3. BACK
4. FRONT
5. ABOVE
6. INCLUSION
7. INSERTION
8. TOUCH
LEFT and RIGHT are self-explanatory. The relations BACK and FRONT should be interpreted as the SO being in the back or front side of the RO in the line that connects the eye of the observer with the RO object. Thus if the SO lays before the RO in this line it is in the BACK of RO the opposite is true if it lays after the RO. ABOVE means, in 3D space, that SO is above the level of the tallest face of RO according to the z-axis. For example in a stack configuration SO is on top of RO. In this case relations TOUCH and ABOVE are active. We assume here that the z-axis is always perpendicular to the workspace and in general it follows the direction of the (negative) gravity vector. The INCLUSION relation means that the SO is (possibly partially) included in the RO. For example RO can be a container box and SO can be a cylinder inside this container. The INSERTION relation captures the relative positioning that a bolt has with two metal plates that firmly secures them in place. See figure 3.1 for explanation. The TOUCH relation describes the situation that the boundaries of two objects are in contact with each other. Further relations, from the set LEFT, RIGHT, FRONT, BACK, ABOVE, are also active in order to show the side of contact between the two objects. Clearly the above model is a first effort to try to capture the spatial relations. It does not contain such detailed information as a CAD-CAM model for a composite object. However, it can easily be expanded with more relations and with more precise metric information, for example by use of the „View Transformation“ matrices returned by the visual system for the various components of the assembly.

Fig. 3.1: The relation of a bolt to the two metal plates that firmly holds them in place is coded by the INSERTION relation in order to distinguish it from the INCLUSION relation that codes the relation between a container and a contained object.

The model is shown in fig 3.2. It receives as input an image from the workspace of the robot where a number of objects might be present. It also receives from the visual system information about each individual object or component of a composite object. This information includes data on the colour, size, “shape” of the simple component or object (which is recognized
atomically) and outputs the vector of activations that includes the eight spatial relations described above. Thus the activations of the components of the vector describe precisely the type of relation that holds between the two objects.

![Diagram of Spatial Relations](image)

**Fig. 3.2:** Model of Spatial Relations. It consists of three major components. The first performs orientation comparison and extracts orientation (alignment) features to be used in the training of the third component which a standard, one hidden layer back-propagation network. The second component is responsible for extracting two main topological features from the 3D contours of the two objects. One feature is the contact of their surfaces, if any, while the other is the inclusion / insertion of one in the other. The features are used for training the third component of the network.

For example, figure 3.3 shows that the red block is LEFT-FRONT of the yellow block, while the green block lies on the top of the yellow one and thus their relation is described as ABOVE-TOUCH. Note that the description that we gave for the configuration of figure 3.3 is the combined result of applying the model of figure 3.2 twice to the pairs of objects (YELLOW, RED) and (YELLOW, GREEN). In both cases, YELLOW is the Reference Object of the relation and it is chosen as such due to the convention we follow that: *the object that is nearest to the centre of the (x, y) plane of the workspace is considered to be the Reference Object*. Note also that the ABOVE-TOUCH relation is qualitatively different from the LEFT-FRONT in the sense that such a relation could be used in the description of a composite object (as is the stack of YELLOW and GREEN) while the other one cannot. In effect, the former is an example of an operator that
constructs whole-part relations for composite objects. The other relations, without the appearance of the TOUCH/INSERTION activations, do not imply any potential composition.

**Fig. 3.3:** Three elementary objects are present in the scene each one of type BLOCK. The RED BLOCK is LEFT-FRONT of the YELLOW BLOCK. The GREEN BLOCK is ABOVE-TOUCH of the YELLOW BLOCK. The YELLOW BLOCK is taken as the Reference Object as it lies closer from all others to the centre of the workspace in the \((x, y)\) plane.

### 3.1.1 Orientation Comparison

The details of the first model component, responsible for the orientation comparison, are shown in figure 3.4.
Fig. 3.4: The component calculates the five major orientations: the three major reference object axes (m1, m2, m3), the proximal and the centre-of-mass orientations. The proximal orientation is defined by the line that connects the two closest surface points each one belonging to the corresponding object. Note that these points might not be unique. The centre-of-mass orientation is given by the direction of the line that connects the corresponding centres-of-mass of the two objects. The orientations are described by the angles (\( \varphi, \theta \)) of the spherical coordinate system in 3D space.

The component on reception of the image and relevant data from the visual system, first it solves a number of computational geometry problems to determine the relevant angles in 3D space which provide the orientation for the three major axes of the reference object as well as of the proximal and centre-of-mass orientations. In each case the 3D orientations are given by the angles (\( \varphi, \theta \)) of the spherical coordinate system in 3D space. In the next layer there are six nodes that correspond to the comparison of the proximal and CoM orientations with the reference object major axes. There are also four nodes that are free to learn any implicit useful orientation in the problem such as the vertical (gravity) direction. These ten nodes are fully connected to the first four feature nodes of the third model component. The ten output nodes are called Gaussian because they use a Gaussian function to calculate the alignment of two different orientations.
For example the first Gaussian node compares the alignment of the RO x-axis with the CoM orientation. The actual calculation is based on the following formula:

$$D_{AB}^2 = (\cos \phi_A \sin \theta_A - \cos \phi_B \sin \theta_B)^2 + (\sin \phi_A \sin \theta_A - \sin \phi_B \sin \theta_B)^2 + (\cos \theta_A - \cos \theta_B)^2$$

(7a)

$$f_A(B) = \exp\left(-\frac{D_{AB}^2}{\sigma_A^2}\right)$$

(7b)

Where $f_A(B)$ is the activation of a Gaussian node that holds the reference orientation (A) which is then compared to orientation (B). $D_{AB}^2$ is the alignment function for the 3D orientations. The parameter $\sigma_A^2$ is a free parameter of the node and can be easily learned. Orientation (A) is assumed fixed and given for the first six nodes of output. It corresponds to one of the major axes of the RO. Orientation (B) then corresponds to the proximal and CoM orientations. The last four nodes can all freely adjust their (A) and (B) orientations to fit the problem at hand. Also note that an orientation X is represented by angles $(\phi_X, \theta_X)$ which correspond to the angle of the x-coordinate axis with the projection of the corresponding 3D line to the (x, y) plane and to the angle of the 3D line with the z-coordinate axis respectively. Figure 3.5 provides a clarification of what is meant by the proximal and CoM orientations.

Fig. 3.5: The first picture shows the proximal (red line, top of blocks) and centre-of-mass (white line, middle of blocks) orientations in the case where the two objects have a LEFT-FRONT relation. In essence the two orientations are almost parallel but vertical to the z-axis. In the second picture we see a stacking scenario (ABOVE-TOUCH relation) where effectively we can only see the centre-of-mass orientation.
(white line) as the proximal orientation is only one pixel wide and thus cannot be observed. Again in this case the orientations are parallel but also parallel to z-axis.

### 3.1.2 Topological Features

The second model component is responsible for detecting two fundamental topological features of the relative position of two objects. The first is the existence or not of (surface) boundary contact. The second is the inclusion / insertion of one object in the other. The existence of one or two of them in parallel can give good information about the relative spatial relation. The component is shown in figure 3.6.

![Fig. 3.6](image)

**Fig. 3.6:** The component consists of a number of maps. The reference boundary mask is a 3D grid representation of activated nodes that are on for all points contained in the boundary of the reference object. The secondary boundary mask is a 3D grid with activated nodes only on for the points that correspond to the surface of the object. These two masks are combined to produce the interior map which corresponds to all 3D grid points that are active in the boundaries of the two objects and in the interior of the reference object. Next follows two features maps which calculate the MAX and AVERAGE functions of their nodes. The feature map nodes receive input from a suitably defined receptive field from the interior map. The output nodes MAX and AVG provide an activation vector which codes the existence or not of a contact or overlap of the objects.

The reference boundary mask collects input from the image and the visual system and produces a 3D grid representation, which codes all robot workspace. Nodes in this grid are active when they belong to the boundary surface of the object. The secondary boundary mask uses the same 3D grid as above and does the same calculation. In a next step the two grids are combined to a new one which is the union of activations of the reference and secondary masks plus the activated interior points of the reference object. All three grids have the same number of neurons in them. The resolution of the grid, i.e. the number of neurons, is selected
appropriately so as to capture enough fine structure from the objects in question and still be the smallest possible number so as to enable efficient computation. The interior map is the main data structure on which two feature maps are evaluated upon. Both of them use the concept of the receptive fields. In our case these fields are quite simple. For a given node, $i$, of a feature map with index vector $(r, j, k)$ we define its receptive field as the one with centre the corresponding neuron $(r, j, k)$ of the interior map and with neighbourhood the six neurons which immediately precede or follow node $i=(r, j, k)$ in all three directions. In other words the neighbourhood consists of the set $\mathcal{N}(i) = \{(r-1,j,k), (r+1,j,k), (r,j-1,k), (r,j+1,k), (r,j,k-1), (r,j,k+1)\}$. Each feature map has a top level node that calculates a given function on the map. The MAX node finds the feature node with the maximum activation in the map:

$$F_{\text{max}} = \max_i F_i$$

(8)

Where the activation of the feature map node $F_i$ is calculated according to the following:

$$F_i = \text{Sigmoid} \left( \sum_{j \in \mathcal{N}(i)} I_{ij} \cdot w_{ij} \right) \cdot I_{SO_i}$$

(9)

Where $\text{sigmoid}(x)$ is the logistic sigmoid function, $\mathcal{N}(i)$ is the neighbourhood of point $i$, $I_{ij}$ is the activation of neuron $j$ in the interior map structure, $w_{ij}$ is the weight connecting node $F_i$ with the node $j$ of the interior map and $I_{SO_i}$ is the corresponding activation of node $i$ in the secondary boundary mask. In effect node $F_i$ is activated only when the corresponding node in the secondary boundary mask is also active. Also note that there are only two weight values for $w_{ij}$. One value is representing the weighting of the node $F_i$ with the center of the receptive field and the other value is common for all other neurons of the set $\mathcal{N}(i)$. Also note that these two values are the same for all nodes $F_i$ in the feature map. During learning only the node with the maximum activation has the right to change these two values which then are copied to all other nodes’ weights.

The top node in the second feature map calculates the AVERAGE of the activation of the map nodes according to the following:

$$F_{\text{avg}} = \frac{\sum_i F_i}{\text{size (Secondary boundary mask)}}$$

(10)
By definition $F_i$ is zero when there is no boundary point at node $i$. The size of the secondary boundary mask is the number of active nodes coding the boundary of the secondary object in the corresponding mask. The MAX and AVG nodes are connected 1-1 to the last two nodes present in the features layer of the third component.

3.1.3 Learning of spatial relations

The third component of the model of figure 3.2 is a standard back-propagation network which consists of three layers. The first, called the features, calculates the values of the input features according to the process described in the previous two components. This layer collects the necessary low-level orientation and topological information that is used for the learning of the top-level spatial relations. It has six neurons. The first three code the alignment information between the reference object major axes with the proximal and CoM orientations. The fourth is a free node coding alignment of the proximal and CoM orientations with implicit orientations in the problem, such as the direction of the vertical axis. The next two nodes copy the values of the MAX and AVERAGE nodes from the second component. All six neurons provide the input training vector. The next layer is a standard hidden layer which tries to build representations of the high-order correlations of the spatial relations nodes with the features vector. The last layer is an activation vector of units that encodes the level of belief that a particular (elementary) spatial relation holds. Values near one indicate the strong recognition of the (elementary) relation, while values near zero indicate the absence of the relation. The vector can represent a complex relation formed by the elementary ones by co-activation of suitable components.

The overall model is a modified back-propagation network which is trained all together following the rules of the standard back-propagation algorithm with the exception of the MAX feature map where there is a “Winner Takes All” type of competition for the right to update the weights and the fact that there are only two weight values which are shared by all neurons of the feature map regarding their receptive fields.

3.2 Simulation Results

The network of figure 3.2 was trained using the objects of year two in DARWIN. Images from the robot’s workspace were used for various configurations, typically presenting simple spatial
relations between distinct objects or a stacking context or a combination of both. Due to the relatively high number of parameters of the model also synthetic data were used to cover workspace configurations that were less frequent or non-existing. Some of these synthetic configurations contained examples for container-contained objects in order to capture the INLCUSION relation. No positive examples were given for the INSERTION relation as this was difficult to be demonstrated with the objects of second year. Thus the effort was given to validate the model and the potential of the approach so as to have a solid basis to extend upon for more complex situations, as for the objects of year three for DARWIN. The development of the model started by developing the 2D case first so as to validate the approach in a simplified setting. The extension of the model in the 3D case used relatively simple objects in order to easily transfer the 2D model to the 3D case. Handling more complex composite object cases requires the development of a computational geometry library, including support for complex mesh geometry, which solves the problem of calculation of proximal and other orientations in an efficient and correct way for the general case, including concave objects, objects with multiple holes and complex assemblies.

Given the development of the model a spatial configuration analysis module was developed which analysed each scene on the basis of pairs of objects; in particular between the reference object and all others in the workspace. Examples of such analysis are shown in figures 3.7 and 3.8 for two different configurations.

![Fig. 3.7: Input scene (left) and spatial analysis output (right). The analysis shows that there is a stack object (point 2) which has to its left and front a red block (point 1).](image)
Note from figure 3.7 that the operator ABOVE-TOUCH implies a potential composite object whose structure can be captured by a tree of suitable spatial operators and components. See for this case figure 3.8 which shows the “shape” representation created for a stack object.

As figure 3.8 shows we can use in principle a tree-like representation for composite objects having as parts the operators and the sub-structures. Such a representation is quite useful in order to be used as a “feature” space by the concept system. A requirement for such a space is to be able to define a similarity function according to the discussion of section 2.3.3 and equations Eq.4 and Eq.5. In our case we can define in multiple ways the similarity function. For example by the degree of matching sub-trees or by assuming a linearised version of the tree and comparing the linearised forms. In the former case we can define the following function:

$$\text{Sim}(a,b) = \frac{\text{# matching relations} + \text{# matching parts per matching relation}}{N \times Q}$$

Where N is the maximum number of spatial relation operators existing in either of the trees (a) or (b), Q is the maximum number of parts (i.e. component objects) existing in either (a) or (b). #Matching relations means the maximum number of common relations that we can find in both trees (a) and (b) when they are traversed with the same order. #Matching parts per matching relation, for a binary spatial operator, means the number of common parts that exist in both
trees in the same argument position. This parameter takes value in the set \{0, 1, 2\} as it is possible to have none, one or two common parts per operator in the corresponding location of each tree. Also note that the number is calculated avoiding double counting in consecutive operators. Even though the space of trees is not strictly speaking metric, thus a proper distance function can be difficult to define, the similarity function in (11) uses topological information in the structure of trees to define similarity of composite objects. Note that the similarity function is always bounded in the interval [0, 1]. An example is given in figure 3.9 together with the calculated value of similarity.

![Figure 3.9](image)

**Fig 3.9:** The two stack objects of figures 3.7 and 3.8 are compared regarding similarity. According to Eq.11 they have one spatial operator in common (ABOVE-TOUCH) in first and second line while the maximum number of operators is \(N=2\) (due to the size of the second stack consisting of 3 objects). They do not have any common argument for their common operator (i.e. for the first line the ‘left’ argument is yellow while the second’s line corresponding value is red. The same applies for their right arguments). Thus they posses zero common arguments (indicating different order of constituent parts) while the maximum number of parts is three (in the second stack). Thus the evaluated similarity is \(\text{Sim}(a,b)=1.0/(2.0*3.0) = 0.167\).
4.0 A representational framework for concepts

This section provides a representational framework for concepts guided by the findings of section 2. The aim is to produce a scheme which can represent not only object concepts (which are the most used representations in DARWIN) but also other entities of more abstract or non-object nature. The key idea is to separate the two main tasks that a concept system does, namely of classification (especially in the case where perceptual features are not discriminatory) and of storing semantic knowledge for the concepts in existence. In section 4.1 we provide the key ideas of the framework and its overall information flow. In section 4.2 we present the overall model and the core competitive network which implements all three representational modes of concepts albeit using different strategies. In the theory-theory mode of representation we also introduce a second model which is suitable for declarative knowledge acquisition. In section 4.3 we produce a concrete representation model and we present some simulation results that support its validity.

4.1 A computational concept representation framework

In this section we will develop a representational framework that corresponds to the properties P1-P6 of section 2.2.1. In section 4.1.1 we discuss the key ideas of the framework, while in sections 4.1.2 we provide the high-level overview of computational / information processing model of the framework. Specific details per representational mode will be given in section 4.2.

4.1.1 Key Ideas

In this section we provide the key ideas that will lead us in the next sub-sections to the representation framework. With every point we provide some brief explanation and/or motivation. The main ideas are the following:

1. *Properties are single domain representations.* Properties constitute an abstract and arbitrary partition of the external world. There are infinite possible partitions. However some of them are considered “natural”. This means that they have the maximum discriminatory power and minimum overlaps among the classified elements of the domain. However, one should not forget the fact that these natural clusters depend
crucially on the sensory (and motor) apparatus that an organism is equipped with. Thus there are no “natural” properties speaking from an objectivist point of view. Also note that properties could be those provided by the perceptual or motor systems or they could be derived through calculation and reasoning (non-sensory set). In any case we assume that a given property relates to only one domain, e.g. colour, texture, weight, etc. Its representation can be selected from any concrete model. Also note that some properties are numerical (continuous or discrete) and some are of nominal (qualitative nature), e.g. Robustness (Break vs. Does-not-Break) or Sex (Male vs. Female). In the first case a typical representation that is used is that of a (growing) self-organizing map, especially for continuous variables. Each single domain property is assumed to define a concrete coordinate/reference system that depends on the problem at hand. E.g. the colour space can be represented with many reference models (RGB, YUV, etc).

Properties are considered as single domain concepts.

2. **Concepts are multi-model representations of properties.** We assume that concepts are a multi-modal representation of properties. As such their representations need to provide a way to support similarity comparison operations on them. This can be achieved in a number of ways. For numerical variables we typically can use (growing) self-organizing maps which are trained on top of the self-organizing maps of the property spaces. Alternatively we can use a representation that builds a Hebbian or other type of association between the concept representation and its constituent properties. This form is natural in a network type of representations. In order to allow a graph model to provide the most active representation we should also introduce two more components. The first is a competition component between values in the same domain. The second is a similarity function which indicates how “close” (or activated) a particular value inside a domain is.

3. **Comparison of instances with concepts in the existence of missing values or partially different properties.** The conceptual representations should have such form so as to allow for the comparison of instances which they do not have either a constant number of properties in each classification session, due to the blockage of one or more property information channels (e.g. being blind-folded and still recognizing a person from his
voice), or they have missing values for a given property. This property should also work well with a mixture of numerical and nominal variables. Much more default values should be provided for the missing values by the concept system with the classification of an instance (following the idea of the taxonomic organization of representations).

4. **Concepts should be learned incrementally.** The model should support the continuous update of the concept’s contents throughout the agent’s lifetime. Thus with every new presentation of an instance potential changes can take place inside the system.

5. **“Summary representations” should be built for concepts.** This follows the idea that prototypes should be built in order to capture the common information that exists in the most members of the class. This also alleviates the need for storing all perceived instances ever encountered in the memory system. With the formation of prototypes the typical representative of the class is created.

6. **In classification of instances to a concept multiple concepts could be provided as a match for boundary or atypical cases.** When an instance is not very close to a prototype, multiple prototypes can be activated albeit with different activation levels. This can implemented both with local as well distributed representations. The latter corresponds to the activation of a linear combination of learned distributed patterns.

7. **Similarity judgments should depend on context.** As noted in section 2 subjects apply different weighting to properties in different reasoning contexts. Thus the formulation should support the application of attention weights to the relevant similarity comparisons.

8. **Similarity should be calculated for both numerical and nominal variables.** This could take many forms, such as distances in suitable metric spaces, implicit models (e.g. ANN), relative ordering relations, etc.

9. **The model should support Theory-theory, Prototype and Exemplar Theories.** As experimental evidence suggests there are strong and weak points in each theory. This
leads us to believe that a model supporting all theories in their strong points will be potentially capable of handling properties P1-P6. A two stages classification process is implemented in this respect. First a new perpect is compared against the prototype representations and if no recognition takes place there classification is attempted in the exemplar level. The exact steps are described in sections 4.2.1-4.2.2.

10. **Concepts with low reward value should be forgotten.** The motivation behind this is that these concepts are probably less frequently encountered (this probably indicates a changing environment) or are less important. If the need arises they could be acquired again. On the other hand concepts that only encountered few times or even once but carry high reward value it should be retained as it they are probably important for the survival of the agent.

11. **Reasoning context should filter the type of information that is returned.** The idea implements the concept of a kaleidoscope. It provides the interface of communication with the reasoning system. If the reasoning system asks for perceptual information this is exactly what it should be returned and not all known information. This fact has a secondary side effect: For a multi-modal representation of the concept we can use as one special domain the “reasoning context”. The values of this domain are all local representations for example having labels as {ISA, IS, CAN, HAS, HOLDS THAT, etc}. The enumerations mean that an ISA context relates to categorization information such as the name of the concept or another suitable reference. The latter can be also part of the multi-modal representation, which can be learned at a later stage. The IS context relates to perceptual information (e.g. colour, size, etc). The CAN context relates to the actions / affordances that they can be supported by the concept. The HAS relates to the parts that the concept consists of. A part could be a link, as a special referential value, to the multi-modal representation of its concept. The HOLDS THAT context provides information for other propositions holding true for the concept. Other reasoning context values can be defined according to the typical uses of information requested out of the concept system.
The main ideas above should provide enough insight as to the formulation of the framework in sections 4.2.1 and 4.2.2.

4.1.2 Overview of the global concept model

This section provides a brief presentation of the overall concept system model. It will explain how the three main theories of concept representation can be combined to provide the overall model. It will also explain the overall information flow that takes place for classification and storage-recall operations.

Figure 4.1 shows the high-level information flow and operations that take place inside the concept system. The system is composed of three spaces of information processing. The first supports the theory-theory (TT), the second the prototypes (P) and the third the exemplar (EX) representations. However for reasons that will be explained shortly the theory-theory and prototypes spaces can be considered as one fused space (TT-P).

When information is received (e.g. from perceptual or reasoning or user interface systems) this is initially directed to the TT-P space for processing. This processing could relate to new information that needs to be classified or an information / knowledge request as to relevant data about a concept. The theory-theory representations are essentially concept maps according to section 2.4, i.e. a set of propositions among concepts and their properties or other concepts. When classification is needed in this space a set of matching concepts is identified through a spreading activation mechanism. Figure 4.12 provides a view of the theory-theory space at an incremental step during learning of new concepts based on workspace input. The model of spreading activation used is the same used in the prototype space and it is described in section 4.2.2. Due to the common model of spreading activation both spaces can be considered fused, thus represented in figure 4.1 by the same level. However, they are really distinct due to the fact that they differ during the recall operation. While the prototype space uses the same activation model for both problems (direct – i.e. linking properties to concepts and inverse – linking concepts to properties / other concepts) the theory-theory space uses for the inverse problem a different model which is described in section 4.2.3 which takes into account, in its structure, the reasoning context which is supplied during a recall request. Thus for recall purposes the answer is returned through the TT space using its concept map knowledge.
structure. For classification purposes the theory-theory and prototypes spaces operate as one and thus classification labels are provided by the fused space during the direct operation.

If classification fails in the TT-P space to find a suitable concept then the classification request is transferred further to the EX space (re-classification attempt). There a match may take place and thus classification information will be returned. If neither of spaces find a suitable match then a new concept is added first in the EX space and then in the TT-P space. In this case the newly created nodes are linked with a weighted connection which is trained with a type of Hebbian learning. With the information/request stream an additional signal is typically provided: that of the reasoning context. This signal provides “instructions” as to which exact data regarding a concept is be returned as an answer (for the case of reasoning system or user requests) or what is the set of “attention weights” which must be used in similarity comparisons between existing concept nodes (prototypes / exemplars) with the incoming percepts for classification operations. The similarity operations can use a default set of weighs (e.g. all set to 1) for all similarity judgments or otherwise.

Fig. 4.1: High-level model of the concept system and related information flows. Three spaces serve as the placeholders for new concepts. The theory-theory space shares the same classification mechanism with
the prototype space thus is considered fused with the latter but differs from it by using a different model for recall operations. Exemplar space uses same mechanisms for classification and recall as prototype space but differs from it in its retaining policy, i.e. exemplars are retained in EX-space based on their reward value while prototypes are kept in P-space based on usage frequency information. P and EX spaces have similar high-level organization but differ in details on how concepts and features values are represented. TT space has the form of a concept map (see section 2.4). Spaces provide either successful classifications or are grown with new concept formation. Recall is always provided by the TT-P space even though classification can successfully take place in the EX one. In the latter case through the connections between theory-theory / prototypes (TT-P) and exemplar (EX) spaces a TT-P concept node is also activated and its properties returned as an answer due to a winner in EX space. See text below for further details.

The way each of EX or P spaces is organized is shown in figure 4.2. There we can see a common structure: Each space is divided in a sub-space of concept nodes (that is grown through new concept formation) and a number of property spaces (i.e. “features”) that represent input modalities to the concept system. Typically the sub-space of the concept nodes is linked with property spaces using a competitive network model which assumes some kind of Winner Takes All (WTA) or similar strategy for development of concept nodes that closely resemble the input percepts. Variations of the WTA theme can be competitive concept nodes with inhibitory weights competing with each other through a spreading activation mechanism. The main differences between the two spaces regarding representation of properties (and their values) and concept nodes are the following:

**Fig. 4.2:** A space (EX or P) typically includes a number of property domains which encapsulate their values in an exclusive set of alternatives. These properties are linked to concept nodes through a set of trainable weights (black lines). Another set of weights (red lines) provides competition among concept nodes in the sub-space of concept nodes. These can be trainable or fixed. In effect the overall processing model is of a
competitive network which implements a WTA or similar strategy providing classification of inputs in an unsupervised way.

- **Property values encoding:** In the EX space the values of a given property are simply stored as they are provided by the input modalities. In the TT-P space the values are mapped to a SOM model which effectively classifies them to a bounded set of values. These values constitute *summary representations* of the property values. The winning SOM nodes (per property) are then linked with trainable weights to the sub-space of concept nodes in the TT-P space.

- **Training strategy of concept nodes:** While both spaces are based on a competitive network model there is a difference in the way the weights, connecting properties to concept nodes, are trained. In the EX space case when a new exemplar is added its weights are trained continually with the same input until effectively resembles it very closely. In the case of TT-P space the corresponding weights are trained only for a single pass (that of the percept presentation) due to the fact that we do not want the concept node that codes for a number of input patterns to resemble anyone very closely, but rather its weights to construct an average representation of the input patterns that activate this node. In this way the node builds a *summary representation* of the relevant patterns it codes for and thus provides a *prototype* for the relevant part of the concept space.

It should be clear from the above presentation how the exemplar and prototype representation strategies are used and why they are complimentary: For every percept that the agent will encounter in its lifetime we do not want to store every input pattern. Thus we need to create prototype representations. However, there are cases which are in the borderline between two prototypes and which are dissimilar enough from the classes’ respective prototypes. These cases will be stored as distinct patterns in the exemplar space. But how the theory-theory view appears in the overall picture? The answer is that this representational strategy can be thought as an extension of the prototype view where additional information pertinent to the concept in question can be stored in the form of propositions (declarative mode vs. self-organising mode of prototypes space). While the prototype strategy typically has as “properties” various perceptual (sensory, motor and reward information) data and creates links between them through the
relevant concept node the theory-theory view identifies a concept present in a “concept map” (see section 2.4) through the activation of not only perceptual information but also through more general relations. These relations typically include but not limited to derived causal structures (e.g. Bayesian causal networks of interacting variables in a given domain or problem) other propositional facts (e.g. statements or beliefs) etc. Thus this view provides an extension of the prototype idea to larger knowledge structures. In some sense the prototype view mainly creates what is called by Ausubel the primary concepts, while the theory-theory approach adds the secondary ones. In principle the former are acquired through direct observation and experimentation in the environment while the latter are provided by user input in propositional form which captures higher-order relationships in the environment (e.g. principles). However both representation strategies can be represented by the same basic competitive network of WTA or similar strategy. What changes is only the scale of the network needed to support the theory-theory approach in contrast to the prototype view. For this reason in the concept system model we fuse both views under one WTA model where we expand the property spaces to include not only perceptual information but also other concept nodes and other properties of non-perceptual nature. At the end the theory-theory mode is the only way we can provide content and meaning to non-perceptual concepts such as “force”, “love” or “electron”. The concept is simply identified by the activation of its high-level relations to other concepts ... but this is structurally the same in the level of a graph model with the competitive WTA network using a spreading activation mechanism.

Before closing this overview we should also note that while the theory-theory representation provides the richest semantic content for any stored concept, the particular implementation through a competitive WTA model is not the best model for storing this content. The reason is that it builds only an average representation (prototype) based on linear correlations and not higher order ones. The latter fact is essential in order to capture correctly exceptions in various reasoning contexts. For this reason we have chosen to approach the inverse problem, of storage and recall of semantic information, through a supervised learning strategy where explicitly we train a network model to code all available information for all concepts in the TT-space given a particular reasoning context for recall. The details of this second model appear in section 4.2.3. It solves the problem on how we get from the knowledge of the concept to its properties; it is based on supervised learning, uses higher-order correlations to construct an internal
representation of the concept and in effect it is a compressed form of Novak’s “concept map” (for all of agent’s knowledge). It can also discover and support implicit hierarchical relations between concepts [36]. On the other hand the model of the competitive WTA network that exists in TT-P and EX spaces is unsupervised, growing and it is based on linear correlations among features to classify the correct concept. It solves the classification problem of how we move from knowledge of properties to the correct concept. The latter is explained in detail in sections 4.2.1-4.2.2.

4.2 Representational Levels, Classification and Storage-Recall of Semantic Knowledge

As it was described in section 4.1 a common competitive network model is used in both exemplar and prototype-theory theory spaces to aid classification of input patterns in an unsupervised way. The particular model we present in next sections 4.2.1-4.2.2 is based on a spreading activation implementation with inhibitory weights among concept nodes (in large distances) and also of excitatory weights among close neighbours. Such structure of weights creates a topological mapping of similar precepts to similar concept nodes. The two sections explain how the classification process is implemented, how the network grows in failure of successful classification and how forgetting is implemented in order to erase concepts that they do not have high-value for the agent.

4.2.1 Exemplar Representation

Let us start with the issue of exemplar representation. The idea that we use is related to the work of James McClelland [7]. Figure 4.3 presents a simplified version of the representation for an exemplar.
Let us consider objects with three features, such as Colour, Shape and Size information. The set of all features is called Feature Space. A node inside any feature set represents an actual measurement (we assume a mixture of non-nominal and nominal variables, e.g. Colour and Shape information are of nominal type while Size is of numerical type). All nodes inside a domain (feature set) are linked with bi-directional links, which are inhibiting each other; in the simplest case they use a common weight, $I$, for these links. Assuming that we have observed Object 1 \{Colour="BLUE", Shape="Stick", Size=20\} we see that there are links from the corresponding feature value nodes to the exemplar node of Object 1. The value “Stick” for the Shape feature can be treated as a linguistic label but this is not necessary; for example it could be a vector of activations in a suitable neural network model for vision. Each of the four semantic nodes (3 for the feature values, 1 for the object) has associated variables of Activation, Input and Effect. The activation is calculated as follows:

\[
Input_i(t) = Sim_i(t) + E \times \sum_j e_{ij} - I \times \sum_j i_{ij} \quad (12)
\]

\[
Effect(t) = (M - Act_i(t)) \times Input_i(t), Input_i(t) \geq 0 \quad (13a)
\]

or

\[
Effect(t) = (Act_i(t) - m) \times Input_i(t), Input_i(t) < 0 \quad (13b)
\]

and,

**Fig. 4.3:** An instantiation of the exemplar space. Three properties are shown: Colour, Shape and Size. The concept nodes appear in the upper part of the space. They connected with inhibitory weights between them.
\[
\frac{d\text{Act}_i(t)}{dt} = \text{Effect}_i(t) - D \ast (\text{Act}_i(t) - R) \quad (14)
\]

where in the above relations Input\(_i\)(t), Effect\(_i\)(t) and Act\(_i\)(t) is the net input, the effect and the activation of each semantic node respectively. Sim\(_i\)(t) represents a similarity contribution arising by incoming percepts. This term in effect calculates a similarity function that is defined in analogy with (3) as:

\[
\text{Sim}_i(t) = A \ast \exp(-c \ast \text{Dist}(AC, i, p(t))) \quad (15)
\]

where in (15) AC corresponds to “attention weights”, A is the magnitude of activation for the similarity function (e.g. with value of A=0.2) and we use a negative exponential, of the Distance function between the node \(i\) and the probe p at time t, to model the similarity function. For our example of figure 4.3 we use the values of AC={1, 1, 1} and c=10 for the “attention weights” and “specificity” respectively.

The distance function in (15) is defined both for numerical as well as nominal (categorical) variables. It takes values in [0, 1] and it is given by (16):

\[
\text{Dist}_{\text{nominal}}(i, j) = \begin{cases} 1, & x \neq y \\
0, & x = y \end{cases} \quad (16a)
\]

And

\[
\text{Dist}_{\text{nominal}}(i, j) = 0, x = y
\]

\[
\text{Dist}_{\text{numerical}}(i, j) = \frac{|i - j|}{\text{range}} \quad (16b)
\]

where, we have defined in (16a) the distance function for nominal and in (16b) for numerical variables respectively. In (16b) the range is defined as range=max-min for all encountered instances seen. For further details for these and other heterogeneous distance function definitions see [8].

Formulae (12)-(14) define a spreading activation mechanism, which is used to calculate the activation of exemplar nodes from presentation of a given percept. The parameters which appear have the following typical values E=0.05 (excitatory weight), I=0.03 (inhibitory weight), M=1 (maximum activation), m=-0.2 (minimum activation), D=0.1 (decay coefficient) and R=0.
(resting level). \(e_j\) and \(i_j\) correspond to excitatory and inhibitory activations from other nodes \(j\) respectively. The activation is called excitatory if it is greater or equal to zero. Otherwise it is inhibitory. Note that we use a single common excitatory or inhibitory weight value. This can be generalised to a full matrix of connectivity among values. The use of excitatory weights together with inhibitory ones in parallel leads to topographic mapping of the input patterns to the exemplar-concept nodes [42].

On the presentation of a new percept we collect its feature vector (which in general is variable in length between observations) and we use as probes its components to the corresponding domain. Using (14) we calculate the effect of the probe in all values inside a domain. Next using (12)-(14) we calculate the activation of all nodes in each domain set and the exemplar set. Before any new activation calculation we initialise the activations of all nodes in random values in the interval \([m, 0]\). The maximum effect that a probe contributes is \(A\), assuming that it exactly coincides with some existing domain value, which we allow it to persist in every iteration. We allow a number of iterations of equations (12)-(14) until the activations reach an equilibrium state. Typical number of iterations is 1000, even though in simulations we have received stable activations even at 100 iterations.

Equations (12)-(14) use three effects to calculate the activation of the node:

1. The effect of the probe; how similar it is to the value represented by a node;
2. The feedback effect from other competitive and cooperating nodes;
3. The exponential self-decay of activation with a constant rate \(D\).

### 4.2.2 Prototype representation

The same spreading activation mechanism, described in 4.1.2, is also in place in the Prototype Level representation. Thus the same underlying model is used. The same parameter values are used as well. As it was explained in 4.1.2 in this level we apply the probes to the (summary) feature values, calculate the effect on each (summary) feature value, which in turn influences the activation of the prototype nodes. At the end some prototype nodes are activated more than others and assuming a decision threshold one returns a set of activated prototype nodes as
the reply of the system to the incoming percept. Figure 4.4 shows an instantiation of the Prototype level corresponding to that of figure 4.3.

![Prototype Level representation.](image)

On the reception of a new percept a classification process takes place. Classification starts at the Prototype Level using the mechanism of equations (12)-(14) to calculate the activations of the prototype nodes. If these activations are less than a threshold, say \( T_1 = 0.2 \), then a second classification takes place in the Exemplar Level in the same manner. If the classification succeeds, then a Concept Update process takes place. If the activations of Exemplars nodes are less than a second threshold, e.g. \( T_2 = 0.1 \), then we enter in a second phase of Concept Formation which is described in the section below. In case that the activations of Exemplar nodes are in \([T_3, T_1]\), where \( T_3 \) is a third threshold, say \( T_3 = 0.15 \), we assume that the percept describes an existing exemplar that needs to be modified slightly or new information to be added to it, such as a new feature. This will be discussed below together with the Update process of the Prototype Level.

**Adding a new Concept**

If our two-stage classification procedure fails to return any activated nodes we proceed on adding the new percept to the Concept System as a new concept. The main idea is that we use a dual representation using both exemplars and prototype nodes. Some percepts will be represented as exemplars; others will only influence the (summary) feature values for the
features of a prototype and they will not have explicit nodes created for them. The set of exemplars that is linked to a prototype node corresponds to the *supporting set* of the concept, which is represented by the prototype node. The whole process consists of four steps. These are described below:

**Step 1 (Exemplar Insertion):** In this step we add in the Exemplar and Feature Spaces nodes that represent the corresponding feature components of the percept as well its related exemplar node. The exemplar node forms links with the newly created feature value nodes assuming a weight of exact same value with the corresponding component of the percept vector for each link (the effect is that the exemplar node will be maximally activate when the same percept will be presented again) [42]. The exemplar and the feature nodes also form links with other exemplar or feature value nodes in their including sets. These links are used for implementing the competition component of the spreading activation mechanism. These links carry a weight of -1. If a component exists in the percept that does not correspond to an existing feature set in the Feature Space, then a new feature set is created where the component value is entered as its first value node. When an exemplar node is inserted an internal variable called LifeTime is initialised as in (17):

$$LifeTime_i(0) = \frac{R - Value_i}{\sum_j R - Value_j}$$  \hspace{1cm} (17)

i.e. it is the relative R-value of the exemplar inside the supporting set. R-Value is the (accumulated over time) reinforcement value of an item. The lifetime variable is used in order to control the time that the exemplar will be stored in the Concept System. Every time that an Exemplar Classification takes place, the winning Exemplar increases its lifetime according to 18a while all other losing exemplars reduce theirs according to 18b:

$$\Delta LifeTime_i(t) = D_1 * LifeTime_i(0), Act_i(t) \geq T_2$$  \hspace{1cm} (18a)

Or

$$\Delta LifeTime(t) = -D_2 * LifeTime(0), Act_i(t) < T_2$$  \hspace{1cm} (18b)

Typical values for $D_1$ and $D_2$ are $D_1 = 0.1$ and for $D_2 = 0.01$
Step2 (Prototype Insertion): In this step we also add nodes in the prototype set and the corresponding summary feature sets. If a new feature set is required then it is created as needed. Links between the prototype node and its corresponding summary feature value nodes are created. Finally competitive links of the new nodes with other nodes in their including sets are also formed. The weights, linking feature values with prototype nodes, are trained (one pass) using a WTA training algorithm.

Step3 (Deriving the Summary Values): Having formed already nodes for the features of an exemplar we need to specify now how these nodes, as well as the corresponding summary value nodes, are initialised. In the Exemplar Representation, the feature value nodes take the value of the corresponding component in the percept. However, in the case of the corresponding summary node, its value is the SOM mapped value most active when it was presented with the feature value in question. We assume that the summary representation for the features values is created through a SOM network for numerical values. In the case of nominal variables, summary values are the same with the corresponding feature values.

Step4 (Linking Exemplars to Concepts): In the final step links are formed between the newly created exemplar and the prototype nodes present in the EX and TT-P spaces. These links use weights that are given by a Hebbian-like rule, more specifically of ARTMAP style (see [9] for details):

\[
\frac{dW_{ep}(t)}{dt} = \gamma \cdot Act_e(t) \cdot (\delta W_{ep}(t) + Act_p(t)) \quad (19)
\]

This rule builds correlations between the exemplar node (e) and the prototype node (p) for all prototype nodes. In (19) \( W_{ep} \) is the weight of the link connecting exemplar node e with prototype node p. \( Act_e \) and \( Act_p \) are the corresponding activations. The parameters \( \gamma \) and \( \delta \) have the values of 0.1 and 0.02 respectively. All weights start with zero value and by use of (19) (see below) evolve over time to stable values (assuming a stationary environment). As the exemplars correspond typically to border-line cases between concept classes, it is reasonable to assume that a given example might belong to a number of classes albeit with a different degree of membership. The weights \( W_{ep} \) define the level of this membership. In a dynamic environment it
is possible that an exemplar can shift from “supporting” a prototype (concept) to another prototype. To calculate the activations in (19) one uses as a probe the feature values corresponding to the exemplar. Activations of all prototype nodes and the exemplar node are then calculated. This process takes place for all exemplars in the Exemplar Space and it is iterated every time there is addition of a new exemplar or there is resolution of the Classification process in the Exemplar stage.

Incremental Change of a Concept

In this sub-section we discuss the related problem of how an existing concept can change given new percepts. There are two principal ways in which this can be accomplished. The first is by change of the summary feature values corresponding to a prototype node. The second relates to the “amendment” of an exemplar node.

Change of Summary Feature Values

If a new percept is successfully classified in the first stage of Prototype representation then a change process in the (summary) feature values of all activated prototypes (above T1) takes place. This process in effect creates the “average” values for the features of the concept. We use an incremental SOM algorithm to achieve this, even though there are other ways to achieve the same effect. However, we believe that a SOM-based approach is both more biologically plausible as well as it provides nice topology preserving properties for the maps of the various feature sets. After the Summary Feature Values are created we also change the weights connecting the summary feature nodes (from the corresponding SOM maps) to the corresponding prototype nodes. Given that the spreading activation mechanism has stabilised we update the “winner” prototype nodes (above threshold T1) using equation (20):

\[
\Delta W_{fp} = \varepsilon \ast (\text{Act}_f - W_{fp}), \text{ if } p \text{ is a “winner”} \quad (20a)
\]

\[
\Delta W_{fp} = 0 \text{ if } p \text{ is not a “winner”} \quad (20b)
\]

Where \(W_{fp}\) is the weight connecting (summary) feature value (f) with prototype node (p). \(\varepsilon\) is a learning rate constant controlling the size of correction step for the winner nodes. \(\text{Act}_f\) is the activation of the feature node (f) in the corresponding SOM map.
Change of Exemplars

It is also possible that a new percept will be successfully recognised in the Exemplar stage and the activations of the resulting exemplars will be in the interval of \([T_3, T_1]\). In this case we assume that the activated exemplars very closely resemble the percept, so the percept should be a “noisy” encoding of the activated exemplar(s). In this case, some adjustment takes place for the feature values of the exemplar(s) based on the newly observed features. Step1 above described the process, covering also the case of novel features. For features that are of numerical nature we assume that the “amended” value is the weighted mean of the two values (of the exemplar’s and the percept’s corresponding values), while for features of nominal nature we assume between the two values the one which has the higher probability.

Controlling the Capacity of the Supporting Sets

Each supporting set has a capacity which is determined dynamically. Capacity is defined as the number of exemplars that are retained in the set. We assume initially that there is a fixed maximum number of exemplars that can be stored dynamically inside the system. Let us call this number \(K\). Assuming that currently in the system exist \(L < K\) exemplars and these are partitioned in \(C\) classes then the capacity for the supporting set of prototype \(p\) is given by (21):

\[
\text{Capacity}_p = K * \frac{\text{Utility}_p}{\sum_p \text{Utility}_p} \tag{21a}
\]

\[
\text{Utility}_p = \sum_j \text{R-Value}_j^p - \text{IE}(P_p) \tag{21b}
\]

i.e. it is the ratio of the set’s Utility, against all other set Utilities. For cases where an exemplar might be linked to more than one prototype we count its contribution to the prototype for which its link has the highest weight. \(\text{R-Value}_j^p\) is the reinforcement value (given externally) of the exemplar \(j\) which belongs to prototype \(p\). IE is the information entropy of the distribution of exemplars belonging to class \(p\), see equation (22) below:

\[
\text{IE}(P) = - \sum_j P_j * \log_2(P_j) \tag{22}
\]
Equation (22) holds strictly for a discrete distribution, \( P_j \), but a straightforward generalisation to an integral can be made for continuous distributions. The usefulness of (22) lies in the fact that it can be applied to both numerical and nominal distributions, while use of standard deviation only works on numerical distributions.

4.2.3 Capturing semantic knowledge

In this section we describe the model of the concept system which implements the storage-recall mechanism of the Theory-theory representation discussed in section 4.1.2. The model not only stores knowledge through perceptual observation but also through declarative means, principally given by a user. It also, following the discussion for the filtering of information necessary due to the reasoning context, explicitly defines the relations of key idea 11 of section 4.1.1. The general form of the model is shown in figure 4.5. It consists of groups of neurons that interact in a specific way. At the input layer there are the nodes that represent a given concept. These nodes have a local representation, i.e. a given node always encodes a specific concept. They are binary variables. In the next layer there are two groups of neurons. The first one is called \( \text{representation} \) and it is directly connected to the concept nodes using full connectivity. This group produces distributed representations for concepts and thus captures the essential high-level correlations that exist in the multi-modal integrative nature of the concept representations. The other group is called \( \text{context} \) and has localist nodes that encode specific relations (i.e. context). For example the CAN context should activate only the nodes of the output layer that are related to applicable actions and affordances of the concept in question. The HAS node should activate nodes regarding structural parts of a complex concept (i.e. a composite object). The ISA relation provides naming information for the concept in question. The IS relation provides perceptual features information such as colour, size, shape and similar attributes. The representation and context groups of neurons are connected to the \( \text{hidden} \) layer group which is responsible for building the non-linear associations between the representation-context nodes with the various attributes. Here the nodes follow a distributed representation. The output layer hosts the \( \text{attributes} \) group which represents all the known attributes for a concept.
Fig. 4.5: The structure of the semantic memory is based on a back-propagation network with a specific architecture. The first layer is provided by localist concept nodes. In the next layer there are two groups of neurons. The first is called representation and creates the non-linear distributed representations for the concept nodes. The second is called context and includes a number of relations typical of the information requests expected from a reasoning system. Both groups connect to the next layer to the hidden neurons group. Finally this connects to the attributes group which uses a localist representation per attribute. Attributes include all knowledge related to a given concept such as category label, actions applicable, perceptual features, constituent parts, etc.

The neurons follow a localist representation. It is clear from the specific structure that as the concept system accumulates knowledge about a concept new nodes for the various attributes (features) are created and associated with the concept in question. The process has been described in section 4.1.1 under the Theory-theory representation and it follows the ideas of Ausubel and Novak of section 2.4. This model can be thought as the inverse of the classification
model of sections 4.2.1-4.2.2. Indeed the latter solves the classification problem and updates features and concept nodes as needed. The former model comes next and associates in an inverse manner the winner concept(s) with their observed or described (declaratively) features. As the model includes all concept nodes and all feature nodes it acts as a global non-linear associative semantic memory. It is shown in [1] that such a model implicitly encodes taxonomic hierarchies of concepts.

The training of the model follows the back-propagation algorithm using an online regime. In other words when a classification task is completed the data of this task (features and recognised or newly created concepts) are used to perform a single iteration training. With the continually incoming information and assuming a stationary environment the model will converge to the correct associations between concepts and their related features. Care should be given for the representation units that learn their activations for a novel concept. In order to explain the process let us assume that the network is already trained to respond correctly to the input signal for a given active concept node and a corresponding active node in the context group. Now assume that the classification process created a new concept. In such a case we have to augment the current semantic memory model with an additional node in the concept group. We do this and we also connect the new (novel) node with full connectivity to the representation group. The question now is how we develop the internal representation for the new concept without disturbing catastrophically the existing knowledge in the network due to other nodes. Remember that the whole network codes for all concepts in parallel and thus its weights carry the semantic knowledge of what is relevant to each concept in a given context.

The answer to the above question is the following: When the new node is introduced we freeze all existing weights on the network except the weights from the novel concept node to the representation group. Given some declarative facts or otherwise, e.g. the name of the class is “Block” or the colour is Red, we apply the current activation pattern of the Representation units to the existing weights of the network and we calculate an output answer. This answer is probably in error and thus we calculate the error and we back-propagate the $\delta$ corrections to each representation node. At this point we change the weights from the novel concept node to the representation group so as to produce a smaller error in the next input-output query. We iterate over the set of declarative statements reducing the output error in each trial and finally we arrive to a stable distributed representation pattern which changes very slowly from this
point forward. At this stage we stop the adaptation of the internal representation. When this adjustment finishes we have a network which added a new concept and still kept intact the semantic knowledge it contained adding in parallel new facts for the novel concept. The end result of the representation process is that the internal representation group captures the semantic similarities of related concepts as similarity in the activation vectors. If the similarity is large enough this could be an indication that a merging process must take place in the concept system. The idea of semantic similarity is shown with some simulation results in section 4.3. In the case where we need to add new facts for an existing concept in the semantic memory we simply modify the overall weights of the network assuming that the appropriate concept node is activated. The two memory updates schemes follow neurobiological findings for a “fast” memory system (the development of an internal representation for a novel concept in our case) and a “slow” one (the overall semantic memory which changes gradually as capturing more complex structure of the environment is achieved).

4.3 Simulation results

In this section we will briefly explain the workings of the model. We will test the system in a number of experiments which provide support for its formulation. In section 4.3.1 we describe the set of objects, used in the DARWIN year two experiments that we used to train and test the model. In addition we explain the relationship with the DARWIN visual system. In section 4.3.2 we provide results that are obtained in a number of experiments using the aforementioned objects.

4.3.1 Visual System input and DARWIN objects

Figure 4.6 shows the processing chain from a visual scene to the output of the concept system for classification tasks.
Fig. 4.6: An input scene in the workspace of the DARWIN robot is captured by the pair of the stereo cameras. Both images are forwarded to the visual system to extract basic object information. The result in turn moves to the scene analysis system presented in section 3 of spatial relations. The output of such processing is then transformed to a set of perceptual vectors which is then fed to the concept system for recognition. The classification output is then forward to other parts of the overall DARWIN system such as the reasoning system.

As it is shown in the figure a visual scene is presented to the pair of stereo cameras overlooking the robot’s workspace. The left and right images are captured and forwarded in the visual system. The output of the system is in terms of:

- The object ID: A unique label to denote the “visual” object class
- The object colour
- The object size (in terms of a 3D bounding rectangle)
- The object position in workspace
- The object pose (relative to a “normal” object view)

The object ID can be associated in a 1-1 way to shape information of the object given that 3D offline models are used to learn to associate the object’s shape from various viewpoints with its unique “visual” class. The ID and colour information is provided as numerical labels, while object size is provided as eight numbers denoting the vertices of a 3D rectangular bounding box. With a
little algebra these vertices can be transformed to a triplet of numbers corresponding to
distances between opposite faces of the bounding boxes along the x, y, z axes direction
providing in this way an implicit indication of object’s size. Object position could be calculated
by the aforementioned 3D bounding box. Finally the object pose is provided as a pose-
translation matrix used for the viewing transformation which is calculated internally by the 3D
visual system.

With this information at hand the next step is the scene analysis along the lines described in
section 3 for the identification of spatial relations among objects in the workspace. The purpose
of this step is to develop composite object representations as discussed there using a tree
representation of nodes corresponding to recognised basic objects by the visual system and
edges corresponding to spatial relation operators. As it is explained in section 3.2 function (11)
can be used to calculate the similarity between composite objects. To allow for ease of
representation both simple and composite objects (e.g. stacks) were represented by a tree of
spatial relations and nodes. In the simple object case, e.g. the RED BLOCK in figure 4.6, this could
be written as RED-ABOVE TOUCH-GROUND (i.e. the red object is based on the ground level). The
two stack object in the same figure is GREEN-ABOVE TOUCH-YELLOW-ABOVE TOUCH-GROUND.
Descriptors of the aforementioned form were then passed to the Percepts preparation stage
where their three attributes (“features”) were coded according to table 4.1. Note that the
various values per feature use a different number of bits to form a distributed representation.

<table>
<thead>
<tr>
<th>S/N</th>
<th>Shape</th>
<th>Colour</th>
<th>Size</th>
<th>Actions</th>
<th>Spatial Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Block (-1,-1)</td>
<td>Red (-1,-1,-1)</td>
<td>Small (-1,-1)</td>
<td>Roll (-1,-1)</td>
<td>8-bit vector</td>
</tr>
<tr>
<td>2</td>
<td>Cylinder (-1,1)</td>
<td>Orange (-1,-1,1)</td>
<td>Medium (-1,1)</td>
<td>Grasp (-1,1)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Mushroom (1,-1)</td>
<td>Yellow (-1,-1,-1)</td>
<td>Large (1,-1)</td>
<td>Assemble (1,-1)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Sphere (1,1)</td>
<td>Green (1,1,1)</td>
<td>Disassemble (1,1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Blue (1,-1,-1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Any (1,1,1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Thus an (atomic) object descriptor and a composite object descriptor would be constructed as a component of a percept vector as shown in figure 4.7. Note that for composite objects (e.g. stacks) their size attribute is considered large if they consist of 3 or more objects, medium if they consist of 2 objects. Simple objects are assumed of being of small size.

**Fig. 4.7:** Object descriptors. Simple objects are encoded using a consecutive list of feature bits. Composite objects compose simple objects using an intermediate 8-bit vector operator describing the corresponding spatial relation between the components. The 8-bit vector is the output of the spatial relations model of section 3. The spatial relations operator is denoted with red colour.

The initial set of percepts/exemplars that was used to populate the concept system is shown in Table 4.2.

**Table 4.2:** Exemplars used in training.

<table>
<thead>
<tr>
<th>S/N</th>
<th>Exemplar</th>
<th>Shape</th>
<th>Colour</th>
<th>Size</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Obj1</td>
<td>CYLINDER</td>
<td>YELLOW</td>
<td>Small</td>
<td>Grasp, Roll</td>
</tr>
<tr>
<td>2</td>
<td>Obj2</td>
<td>BALL (Sphere)</td>
<td>BLUE</td>
<td>Small</td>
<td>Grasp, Roll</td>
</tr>
<tr>
<td>3</td>
<td>Obj3</td>
<td>PARALLELEPIPED</td>
<td>GREEN</td>
<td>Small</td>
<td>Grasp</td>
</tr>
<tr>
<td>4</td>
<td>Obj4</td>
<td>PARALLELEPIPED</td>
<td>YELLOW</td>
<td>Small</td>
<td>Grasp</td>
</tr>
<tr>
<td>5</td>
<td>Obj5</td>
<td>PARALLELEPIPED</td>
<td>RED</td>
<td>Small</td>
<td>Grasp</td>
</tr>
<tr>
<td>6</td>
<td>Obj6</td>
<td>MUSHROOM</td>
<td>GREEN</td>
<td>Small</td>
<td>Grasp</td>
</tr>
<tr>
<td>7</td>
<td>Obj7</td>
<td>STACK (Y-G)</td>
<td>MIXED</td>
<td>Medium</td>
<td>Grasp, Assemble, Disassemble</td>
</tr>
<tr>
<td>8</td>
<td>Obj8</td>
<td>STACK (R-Y-G)</td>
<td>MIXED</td>
<td>Large</td>
<td>Grasp,</td>
</tr>
</tbody>
</table>
In the above table we use a set of elementary objects (6) as well of composite ones (2). The latter represent stacks. For each object we provide values for its elementary features. Note that we combine shape, colour and size information to create the composite descriptors as a high-level feature. Thus the percept vector for the simulations in the next section contains only two components: Composite-object-descriptor and action descriptor. The action feature can take multiple values for the elementary actions such as grasp, roll the object and assemble/disassemble a composite object. The latter two actions do not hold for elementary objects. The above combinations of feature values define a set of fourteen unique object instances which can be used to train the concept system.

4.3.2 Experiments in DARWIN

In this section we discuss the performance of the concept system with the objects of the second year from DARWIN. Typical objects and configurations were presented in section 3.2 and are shown in figure 4.8 for ease of reference. The overall set includes blocks (parallelepiped) of three colours, a mushroom kind of object of green colour and a yellow cylinder. The objects were coded according to three perceptual properties {Colour, Shape, Size}, the affordances they support and their class label. Colour, Shape and Size are combined to define high-order object descriptors for elementary and composite objects. In the summary level representations of the TT-P space SOM maps were used with nine elements each. For the perceptual features, the encodings used were those in table 4.1. To be able to handle both composite and elementary objects we used the assumption that each object can be coded using the tree representation of composite objects even in the case of the elementary objects using the convention that an elementary object can be considered to have an ABOVE-TOUCH relation with the GROUND object which represents the floor of the workspace.
Fig 4.8: Objects in year two of DARWIN used for testing classification performance in the concept system.

The SOM map used to code the composite object descriptor feature used a distance sub-function that, for these simulations, is defined as the Hamming distance function between the corresponding components of the linearised form of the tree representation. This has provided a straightforward application of the SOM algorithm in comparison to function (11) which provides a definition of similarity for a composite objects’ shape feature.

With the above assumptions and clarifications the classification process proceeds as it was explained in sections 4.2.1-4.2.2. The percepts included only two components (composite object
descriptor and action descriptor), thus only two SOM maps were present for in the TT-P space to code for the corresponding summary feature values.

We show in figure 4.9 the produced SOM map for the composite-object-descriptor feature which was discussed above. As we observe lighter colours indicate the existence of a stack with increased number of BLOCK components participating in it. As the colours become darker we move from stacks with high number of BLOCKS into stacks with smaller number of BLOCKS and/or MUSHROOM and CYLINDER components participating in them or single BLOCK objects. The darkest colours indicate the existence of single MUSHROOM or CYLINDER objects.

![Fig. 4.9: The resulting SOM map for the unified composite-object-descriptor representation produced for elementary and composite objects. Lighter colours indicate a stack with an increased number of BLOCK components. Medium dark colours indicate either elementary BLOCK objects or stacks with a BLOCK and other shape components. Darkest colours indicate the existence of the elementary MUSHROOM and CYLINDER objects.](image)

Figure 4.10 shows a sequence of snapshots from the training of the TT-P space. Here we have projected the percept vectors and the corresponding developed prototypes into a 2D space using a multi-dimensional scaling technique which preserves the topological relations among the objects. To train the system we have used the set of fourteen objects of table 4.2 for 200 sessions of presentation. In each session the order of appearance of a given object from table 4.2 was random. The learning rate in Eq.20 was set to $\varepsilon = 0.1$, for the winner nodes. The “attention” weights of Eq.15 were set so as to bias similarity more towards the composite-object-descriptor comparisons rather than the possible actions. The specificity was set to $c=1.5$, while the $T1$, $T2$ and $T3$ were set as described in section 4.0, i.e. $T1=0.4$, $T2=0.1$ and $T3=0.2$. The parameters of Eqs12-14 are set as described in the text following the equations.
Note that due to the fact that the Y2-Y3 DARWIN object concepts were mainly defined by their perceptual features (rather than by more abstract declarative knowledge) figure 4.10 focuses on presenting the incremental self-organisation and growing of the prototype and exemplar spaces inside the concept system. The theory-theory view while exists in parallel (we see a snapshot in figure 4.12) it is not actually very different from the information contained in P-, EX-spaces, thus we will limit the discussion of the results to these two spaces.

Figure 4.10 consists of a sequence of images corresponding to different training steps inside the system. To the left image side the P-space is presented with blue coloured prototypes while the descriptors of objects of table 4.2 are shown in black letters. To the right side the EX-space is presented with the same object descriptors and the exemplars developed are in red colour. The fourteen instance objects are represented through the numbers 1-14 in black colour. Also note that for ease of tracking in the prototype space we draw connecting lines between prototypes and the corresponding percepts they code for. For exemplars this is not the case as the exemplars always code a single percept closely.

The scenario that was followed in this simulation was as follows:

1. All fourteen prototypes were presented in each training cycle, for the first 100 cycles, but the order of presentation was random. For the next 100 cycles only the prototypes \{1, 13, 14\} were present. This was in order to test the forgetting abilities of the system.
2. The numbers present in the images correspond to the following classes of objects:
   a. 1-2: Cylinders with roll (1) and grasp (2) properties
   b. 3-4: Balls with grasp (3) and roll (4) properties
   c. 5-6-7: Blocks
   d. 8: Mushroom
   e. 9-11: Stacks of medium size (2 objects) with different grasp (9), assemble (10), disassemble (11) properties
   f. 12-14: Stacks of large size (3 objects) with the corresponding properties as above
Fig. 4.10: Stages in learning as the system discovers new groupings in data in prototype space. Initial condition is shown in the first image of the sequence. Each image has to the left the view of the prototype space and to the right the view of the exemplar space. Left of each image is the training iteration number. The system starts with only one prototype/exemplar to code the whole space. However, this is not very discriminating and another prototype is added just after two steps. At step five the system has already four. Eventually the system covers the space properly at which stage we start the second phase of the simulation which keeps alive only three of the original prototypes (step 101). After this the system starts forgetting prototypes/exemplars that are not supported any more from data. Note that the number of exemplars and prototypes in corresponding spaces could be different as the former uses a retaining policy based on reward value and the latter on frequency of appearance. This setup creates advantages, e.g. in case of remembering the ‘grandmother’s’ face even though we take a long time to meet her again, assuming that ‘grandmother’ is coded as an important exemplar.

We observe that the system starts with a single prototype (step=1) and when in next steps fails to classify correctly the new data, an exemplar is created in exemplar space and then in turn it creates a prototype vector in TT-P space. So at step=5 we have already four prototypes. At step=10 we have further differentiation but we have effectively covered the space of percepts. The only erroneous condition concerns prototype #6 which effectively does not win any competition in order to code for data. This is shown with lack of any link connecting prototype #6 to any of the data points. At step=60 this problem has been corrected, by eliminating prototype #6, but another one remains. This is that prototype #5 is coding for both the mushroom object and a block (i.e. object 5, 8). However, with more presentations, eventually
this problem will also be corrected at step=90. There prototype #4 codes correctly for all three block instances, while #5 starts moving towards the only mushroom instance. Regarding the other prototypes present in the system we see that all code correctly their classes, specially #6 and #7 which code for medium and large size stack objects. The only other problem remaining is that #1 and #2 prototypes code single instances of cylinders. This is generally not a problem as with increasing presentation time one of them (#2) will code for both cylinder instances (1,2) and #1 will be eliminated eventually. However, this last development occurred in another simulation sequence which is not shown here. Here we selected to show the evolution of the system as the number of prototypes presented is reduced in order to show the forgetting abilities of the system. Indeed after step=100 only three of the original prototypes are kept for presentation. These are {1, 13, 14}. The image of step=101 shows that only three of the original instances survive. However, the prototypes still remain, even though completely unsupported from data except of the case for the #1 and #7. What we see next in steps=110-173 is the gradual elimination of unsupported prototypes until only the ones with data support to remain. Also note that in the whole sequence the exemplars quite early were eliminated as they actual served no useful purpose given their initial reward value (which was the default value of 1.0). Given that the data space was effectively classified by prototypes only there was not any reason why the initial exemplars to remain and thus the latter were gradually eliminated. On the other hand if one needs to keep some of them around needs only to assign appropriately high reward values for the application domain.

It is clear from the simulations that the interaction of the specificity value (c) with the lowest level of recognition (T2) controls the number of prototype/exemplar vectors that they will be created. If specificity value is small for a given T2 value more prototypes are created. If specificity increases less are present. In effect the c/T2 variables control the resolution of the system. In the one extreme case all data could be coded by a single value, actually capturing no real differentiating information in the percepts. On the other extreme the prototypes could become the same number as the exemplars in effect cancelling any information compression and generalization properties of the system in this trivial case. More research is needed to determine a generic policy that can be possibly applied in a domain agnostic way as to how best determine values of c/T2 in order to control the targeted level of resolution for the system.
Initial strategy used is the minimal non-overlapping partition of prototypes that leaves no data point excluded.

The above simulation, others with different scenarios, experiments on the lab and also applying the concept system to non-DARWIN data show strong support for the validity of the concept system’s architecture. As further evidence in the next figure, fig. 4.11, we see the application of the concept system to a set of 2D-projected data originated from marketing research data.
Fig. 4.11: Learning stages of the concept system in marketing data. Two natural clusters exist in the data and the system through a series of progressive steps eventually maps the problem domain correctly. The same concept system configuration was used as in the case of the DARWIN data. Default parameter values seem to be valid in multiple application domains.

It is worthwhile to also note that the values for the various system parameters that we used were the same for both data sets. Thus the problem of setting algorithm parameters per domain application does not seem to be severe. The default values used seem to achieve good performance on many different types of data sets. Also note that there is no direct way to say to the concept system how many clusters of data exist in the input space. The system itself discovers the natural clustering relations, even in a non-stationary environment, through a set of internal self-organisation processes. The initial strategy of setting the specificity / recognition level variables work for the tested cases, even though more research is needed to establish reliable conclusions.

Even though in the simulation results we focussed our attention to the prototypes and exemplar spaces, for reasons presented above, nevertheless we provide in figure 4.12 a snapshot of the learned representations in the theory-theory space. In effect it follows closely what is actually coded in a compressed form in the corresponding prototype vectors. Thus, in the case of DARWIN, it does not offer much difference in comparison with the prototype space for Y2-Y3 data. However, if a user starts training the DARWIN robot providing declarative knowledge then this representation is the most suitable layer of the architecture to code these higher level
relations in the agent’s environment. In classification operations the TT-space will provide more complete and faster recognition than that of the P- and EX-spaces for recognition of a concept through its high-level relations.

**Fig. 4.12:** Incremental formation of concept map produced during training on input provided after workspace scene analysis (effectively the exemplars of table 4.2). Circular gray nodes denote the concepts (exemplars), darker blue nodes are class labels, lighter blue nodes are values of perceptual properties and yellowish nodes are action/affordances properties of object classes in question. Links are relations that correlate concept nodes with their properties thus forming propositions which represent the overall knowledge structure [41].
Before concluding the section it is worth mentioning that using the model of section 4.2.3 we have developed internal representations for the objects discussed here according to the process described there. The end result of developing the internal representations shows indeed that the similarity of internal representations is isomorphic to the semantic similarity that exists for the various concept classes as expected. Figure 4.13 shows internal representation vectors for the various cases discussed in figure 4.8.

In the model we used six neurons to encode the internal representation of concepts.

As it can be seen from the figure all objects have very similar activations in the corresponding neurons. This is explained by the fact that the elementary objects are basically the same from the point of view of attributes that they possess in common. Except of colour and shape components all of them share the same size and the same affordances (can be grasped, and possibly rolled) even though they differ in the class label as well. Stacks resemble single block objects as basically they are the same with them with the exception of the number of participating blocks in the composite object which further reinforces the activation.
5.0 Brain-guided organization of Concepts in DARWIN

Recent developments related to the organization of semantic knowledge in the brain (see [11] [12], [13] for detailed reviews) provide crucial insights that help to constrain computational architectures modelling the development and organization of concepts in embodied cognitive agents. The main finding emerging from these results is that conceptual information is grounded in a “distributed fashion” in “property specific” cortical networks that directly support perception and action (and that were active during learning). Same set of cortical areas are known to be active both during real perception, imagination and lexical processing. It is also established that “retrieval” or reactivation of the neural representation can be triggered based on partial cues coming from “multiple modalities”: for example sound of a hammer retro activates its shape representation [14, 13], presence of a real object (banana) or a 2D picture of it can still activate the complete network associated with the object (and that was active during learning of it in the first place). The results indicate that while there is a fine level of “functional segregation” in the higher level cortical areas processing sensorimotor information, there is also an underlying cortical dynamics that facilitates ‘cross modal, top down and bottom up’ activation of these areas. “Higher level” is emphasized because there is reason to believe that both early stages of perception (lower level color, shape processing) and late stages of action (like, muscle activity) should not be involved in embodied simulation. Otherwise it would become impossible to distinguish simulation from reality (and we believe retaining this distinction has advantages in computational terms too). There is evidence supporting this viewpoint both from motor [15] and perceptual studies [16]. In the sections that follow, we attempt to transform the findings from neuroscience into a possible computational framework for organization of concepts in DARWIN robots (and conduct experiments to understand the resulting benefits in terms of the inferential capabilities of the robot). Section 5.1 describes representation and cumulative learning of object concepts in DARWIN with several experimental results. Section 5.2 extends this framework and describes how object, action and body schema are connected in the DARWIN architecture.
5.1 Distributed “property specific” organization of object concepts:
Learning, reasoning and growing as it takes place

Figure 5.1 shows a block diagram that captures the building blocks and information flows that leads to distributed representation/learning of object concepts. We briefly summarize the details below.

*The sensory streams:* At the bottom is the DARWIN sensory layer that includes the sensors and associated lower level communication protocols and algorithms to analyze properties of the objects, mainly color, shape and size. The color of objects is analyzed by a color segmentation module (developed by CVUT: D3.1) based on a recent approach using Markov random fields [17]. This returns a triad of RGB values which forms the input to the color SOM. At the level of concept system, information related to object shape is passed as 120 bit vector unique for each shape (the actual shape recognition performed by the computational framework provided by CVUT: D3.4). In this way, the complexity of shape analysis is abstracted from the concept system. Size related information is organized into two different maps one coding for magnitude (the maximum length of the object across any axes in Cartesian space: say S1) and proportion (i.e. the ratio of the maximum length with respect to lengths in the other two axes, say S2). S3 relates to orientation that is not a property of the object itself but rather is relative to the frame of reference of the observer. This kind of organization of size related information is partly inspired by recent evidence related to representation of magnitude in the parietal cortex [18]. There are several advantages of this scheme in terms of inferring what can be done with different objects that may be indistinguishable through color or shape (for example consider a green cube and a green stick: both have same shape and color, what distinguishes them is the abstract magnitude and proportion: the former can be used to build a stack the latter as a tool to pull an unreachable reward). Word information is the input directly coming from the teacher. Infants often learn to associate “words” with objects by learning in a social environment and interacting with the parent/teacher. It is further possible to exploit compositionality in the domain of words. For example consider an example of a “black apple”, even though we may have never encountered such an object, we can easily “imagine” what it should be and this should activate “top down” higher level areas processing color and shape and not just words as is known from several studies in brain imaging [13]. At present “word” related inputs are
entered by the teacher using the keyboard and converted into vectors on the basis of letter usage frequencies in English language as is done in [19]. In the present system, a sequence of maximum three words or lesser describing the object (size-color-shape, for example “small red cube”) are considered, the resulting individual activity superimposed to get the final activations in the word SOM. From an “application perspective” incorporation of little linguistics (that is grounded in sensorimotor experience of the learner) endows the architecture with a measure of user friendliness. In the future we look forward to replace the input modality from the “keyboard” to a direct auditory channel (along the lines of work done in the EU funded FP7 CHRIS project or other available speech analysis software).

5.1.1 Learning and self organization of color, words and shape related information

The information coming from the DARWIN sensory layer is projected bottom up to a set of growing self organizing maps (SOM) learning and representing object properties at a conceptual level. The first level neural connectivity between the sensory layer to property specific SOM’s are learnt using basic SOM procedure [20, 21]. As we go higher up in the hierarchy (figure 5.1), the representations become more multimodal and there is greater integration of information coming from multiple SOM maps (in layer 1). Here we need go beyond the standard self organizing maps and introduce some novel concepts for learning “layer 1 SOM to hub” (and higher up “hub to hub” neural connectivity). In general, hubs are also self organizing maps but higher up in the processing hierarchy and serve two main purposes:

1) Facilitate multimodal integration of information arriving bottom up (from the sensory streams through layer 1 SOM’s)
2) Enable both “top down” as well as “cross modal” activation of various “property specific” maps during reasoning and resolution of contradictions.

As seen in figure 5.1, we distinguish between two kinds of hubs: “provincial hubs” that integrate neural activity coming from small sets of lower level SOM’s and “connector hubs” that integrate information coming from provincial hubs. An analogy may be that of a team leader (who works on a specific problem with a group of 3-4 students) and the director of a department (who is the face of the organization for the external world). The connectivity between property specific maps and hubs are developed using three additional rules as described below:
1. **Preferential Attachment**: This idea is simple and just means that there is a tendency of individuals to “preferentially” connect to other highly connected “individuals” (instead of randomly connecting to anyone in the network). This has the net effect of reducing path length between any two individuals in a large scale complex network. It is well known from network theory that preferential attachment gives rise to growing scale free networks with small world properties [22], a feature prevalent in many real world systems. In the initial attempts to create growing networks with small world properties, preferential attachment of new nodes was directed towards existing “highly connected” individuals with greater “nodal degree”, hence modelling a kind of “rich get richer” phenomenon. In this case, the previously existing nodes (or senior ones) have a clear advantage over new comers. If this is the case then how do new comers make it in a world where “rich get
 richer”? Realizing this issue, Barabasi [24] proposed a measure called as “fitness-connectivity” index hence combining “fit gets richer” with “rich get richer” to create growing networks. “Fitness” is generally “context dependent” can be attributed to different factors based on the network in question (power grids, internet, air transport). Considering that ‘space and wiring’ constraints play a crucial role in the emerging connectivity of the brain, we decided to have a gradient of “fitness” so as to promote layer 1 neural SOM’s to preferentially connect to “provincial hubs” (and “provincial hubs” to “connector hubs”). In the biological case, we believe it is plausible that evolutionary pressures and genetic factors may play a role determining “fitness” of cortical areas to promote preferential attachment.

2. Temporal coincidence: This simply means that if neurons in different self organizing maps are concurrently active (within a temporal window), then they get connected to each other (not directly) but through the “provincial hub” in their territory. Note that, being connected through the provincial hub (and not directly) ensures that there is both functional segregation (between different neural maps) and at the same time global integration. An analogy is two doctoral students working on their own problems, collaborating at instances and connected through a team leader. There is close contact at the same time a level of local functional autonomy.

3. Dual Dyad connectivity: If there are 3 nodes then there are 13 ways to connect them (12 of which are shown the left panel of figure 5.1). C. Elegans is a tiny worm measuring about 1mm whose brain (with about 302 neurons) has been exquisitely studied for almost 3 decades. Way back in 1985 the overabundance of “triangular sub circuits” of a particular type called as “dual dyad” (highlighted in figure 5.1) in the brain of C. Elegans was noted by [24] and this has been confirmed in several other subsequent studies. More recently analysis of the cat and macaque cortex has also revealed that “dual dyad” connectivity is found in significantly high proportions [25]. This implies that such kind of connectivity comes with advantages (hence being retained by evolution). Guided by these studies, while connecting neurons belonging to different neural SOM’s, we have retained the “Dual dyad” type connectivity. The computational gains of having such reciprocal connectivity between multiple maps will be demonstrated gradually in various sections.
Figure 5.2 (right panel) shows activity in the color, word, shape and provincial hub maps while learning the first steps of associating names of different objects (given by the user) with its perceptual properties (color and shape processed bottom up through the sensory channels). The left panel results will be elaborated in the next subsection after describing global network dynamics of the “small world”. In this subsection, we merely focus on how the connectivity between various neural maps are developed (the connectivity relates to “color”, “word”, “shape” and provincial hub).

Fig. 5.2: (left panel) shows neural activity in the self organizing maps related to color, word, shape and provincial hub while learning a simple “small world network” that brings together all these functionally segregated neural maps (all driven bottom up by sensory channel) into a globally integrated system (at the level of provincial hub: which is analogous to a team leader working with 3 graduate students). Five different cases are shown in the left panel. In every case, the robot is presented with a novel object and coincident with a word sequence (provided by the teacher). Neurons in different property specific maps that have sensory weights closest to the incoming input signal start representing these signals (with their sensory weights gradually adapted as in standard SOM’s). At the same time, connections are developed between ‘property specific’ SOMs and the provincial hub due to preferential attachment and temporal coincidence (for example, winner the color SOM connects to the winner in the word SOM through the
provincial hub by means of dual dyad connectivity pattern). In the left panel, the activations in the word and provincial hub are shown twice as they correspond to activations in response to individual components (color and shape). The net activation can be considered as the superposition of activations resulting from individual components (like in the right panel). Fig. 5.2 (right panel): Shows the local inferential powers of even this “small patch” in the DARWIN architecture. When someone else mentions the word “red horse” or grasp a “black apple”, most of us may be able to anticipate what this new sequence of words may refer to. If a black apple is eventually kept in front of us, most of us would even grasp it because we can anticipate top down what a novel object “could be” and if bottom up sensory input activates the neural maps exactly in the same way as top down, we can infer this object is indeed the “black apple”. Action networks could be triggered to initiate the action, though the goal was unheard of. The same scenario is replicated on the robot: the user inputs a new word sequence, we observe how activity in the “word map” retroactivities gradually in time, the complete network (in a very different way from what was learnt: right panel). If a blue cube is indeed kept in front of the robot, “top down” and “bottom up” activity will resonate allowing the robot to infer that the new object is indeed the blue cube, that it has been commanded to grasp.

**Fig. 5.2.b:** Left and middle panels show the teacher presenting yellow cylinder and green cube during the learning phase and right panel shows the teacher presenting a novel object i.e. the blue cube that the robot both recognizes visually and is able to come up with the correct name due to cross modal activations in the neural maps.

Let N be the number of neurons in any SOM and S be the dimensionality of the bottom up input feeding the map. Then the connectivity matrix has a dimensionality of N x S. Since we are dealing with multiple maps here, for clarity we address N_C, N_S, N_W and N_PH as the number of neurons in the color, shape, word and provincial hub respectively. Since color, word and shape SOM activity forms the bottom up input to the provincial hub, the connectivity matrix of the provincial hub has a dimensionality of N_PH x (N_C + N_S + N_W). Since all SOMs are growing, N itself is a function of time and experience that the robot acquires. For the illustration purposes in figure
5.2 activity of 9,9,30 and 36 neurons in the color, shape, word and provincial hub maps are shown. Five different cases are shown in different rows, in each case the robot is presented with a new object followed by the linguistic input of what it is from the teacher. In the first case the robot is presented with a yellow cylinder. Color and shape is analyzed bottom up through the sensory layer and feed the respective SOMs with sensory vectors $S_C$ and $S_V$ respectively. In the same temporal window of integration, the teacher inputs the word sequence “yellow cylinder”. The two words are inputted in a sequence and the activity in the word SOM and provincial hub in response to individual components (in this case word 1 describing color and word 2 describing shape) is shown separately in figure 5.2 (left panel). The net activation in the word SOM and provincial hub can be visualized as the superposition of the individual activations (like in figure 5.2 right panel). The different sensory streams activate bottom up the various layer 1 neural maps that initially have randomly initialized connectivity matrix. Layer 1 maps are trained in parallel using the standard SOM procedure that is fairly standard and discussed in detail in numerous references see [21, 26]. In short, this consists basically of two steps:

1) Finding the neuron ‘$i$’ that shows maximum activity for the observed sensory stimulus $S^t$ at time $t$. This also implies that neuron ‘$i$’ sensory weights $s_i$ such that $||s_i-S^t||^2$ has the smallest value, among all neurons existing in the respective SOM at that instance of time.;

2) Adapting the sensory weights of the winner in a Hebbian fashion by bringing the sensory weights $s_i$ of the winner “$i$” closer to the stimulus $S^t$. This simply has the effect that in future instances the neuron “$i$” actively codes for the particular sensory stimulus $S^t$. In this way neurons in different property specific maps of layer 1 that have sensory weights closest to the incoming input sensory vector start representing these signals.

The net activity in the color, word and shape SOM’s forms the bottom up input to the provincial hub. The connectivity is of dual dyad type, and weights are adjusted in two identical steps one relating to “color-word” association and other related to “shape-word” association. This is because the teacher is inputting a sequence of two words, the first related to color and second related to shape (of course nothing stops from training the maps separately in a distributed organization scheme, for example while showing a yellow paper and uttering the word “yellow”, shape map is just switched off and learning takes place between hub and color SOM’s). However
we chose to train them together because color and shape maps do not generally interfere with each other. Haptics has just recently been incorporated in iCub humanoid (see EU funded Roboskin project for details) and work is ongoing to exploit this modality, but at present vision is the main source of sensory information. The learning rule to connect layer 1 SOM’s with the provincial hub is as follows:

‘if neuron “i” and neuron “j” winning in the color and word SOM’s respectively manage to activate neuron “k” in the provincial hub , make $W_{ik} = 1$ and $W_{jk} = 1$. This has a net effect of enabling neuron “k”, “i” and “j” in three different SOM’s (operating on their own local sensory streams) to retro activate each other in “bottom up”, “top down” and “cross modal” fashion. The same applies to adjusting connectivity between shape, word and hub SOM’s.’ The internal weights of the provincial hub can either have random initialization or a winner “k” can be randomly chosen from the subset of neurons in the provincial hub that have internal weights zero. The net effect is that in both cases there is some neuron in the ‘provincial hub’ that responds to activity in two different SOM’s processing different sensory streams. Activity in any map can gradually trigger the whole network hence enabling “pattern completion” from a partial cue.

We start with 5 objects (of different colors and shapes associated with their names) that are taught to the robot. The activity in various SOM is shown in figure 5.2. The activity in the “word” and “provincial hub” maps are shown twice for clarity, because the teacher input consists of a sequence of two words. As we can see, in every layer 1 SOM, different neural units start learning and representing different sensory stimulus. In the future if similar stimulus is projected bottom up, then the neuron coding for it is reactivated. For example as seen in figure 5.3 (right panel, row 1), showing just a red paper to the robot activates the neuron coding for “red” sensory stimulus in the color SOM processed bottom up through vision (and experienced first while the robot was presented with the red pyramid). At the same time, activity in the color, shape and word SOMs are integrated at the level of the provincial hub by means of the “dual dyad” connectivity pattern. This also implies that showing a “red paper” to the robot should “cross modally” activate the word representation “red” in the word SOM, even though in this case there is no word input from the teacher. This is indeed the case. In other words, just perception of color “bottom up” is sufficient to retroactivate the global network learnt during past
experience, but at the same time just associated with the particular “partial cue” perceived in the present context (in this case there is no activation in shape map). This behaviour is very common in infants (show them a “dog” for example and say the word “bow bow”, next time the child sees a dog, we often see it playfully pointing to it with the word “bow bow”). Studies in functional imaging go even further providing evidence that even if it was a toy dog, a real dog or a cartoon or just the word “bow bow” should activate the global network as was experienced during learning [27]. To further understand how this “cross modal” and “top down” retroactivation takes place when the network is triggered with a partial cue from the environment, we look into the global dynamics of the “small world”, which is the topic of discussion in the next section.

5.1.2 Network dynamics, Pattern completion and Multimodality

In the proposed distributed small world organization, even the simple “color-world-shape” network consisting of just four neural maps is endowed with its own “local” ability to both “reason” in novel situations, grow and resolve contradictions that may arise between what the system anticipates “top down” and what actually activates the system “bottom up”. To achieve this objective the small world network has to be complemented with an equally powerful dynamics that allows neural activity in one map to retroactivate other relevant networks in “top down”, “bottom up” and “cross modal” fashion. The network dynamics builds upon the idea of neural fields [28] and supplements it with novel concepts like the introduction of the bifurcation parameter [29] that both brings in computational advantages and is biological plausible (as will be discussed later). Let \( h_i \) be the activity of the \( i \)th neuron in the provincial hub and \( x_{prop} \) be the activity of a neuron in any of the property specific SOM’s connected to the provincial hub (in this case color, word and shape SOMs). Let \( W_{prop, hub} \) encode the connections between the property specific maps and the provincial hub. Basically, \( W_{prop, hub} \) is a \( N_{Prop} \times (N_c+N_w+N_s) \) matrix learnt as explained in the previous section. Its transpose encodes backward connectivity from hub to individual maps. The network dynamics of hub neurons and neurons in the property specific maps are governed by equations 23 and 24 respectively:

\[
\tau_{hub} \frac{dh_i}{dt} = -h_i + (1-\beta) \sum_{i,j} (W_{prop, hub} X_{prop} + \beta (Topdown))
\]

(23 and 24)
\[ \tau_{\text{prop}} \dot{x}_{\text{prop}} = -x_{\text{prop}} + (1 - \beta)S_{\text{prop}} + \beta \sum_{i,j} (W_{\text{hub}, \text{prop}} h_{\text{hub}}) \]

Where,

\[ S_{\text{prop}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - S)^2}{2\sigma^2}} \]

The instantaneous activation of any neuron in the hub or the property specific maps is governed by 3 different components: The first term induces an exponential relaxation to the dynamics. The second term is the net feed forward (or alternatively bottom up) input. Since property specific maps are inputs to the provincial hub, activity of neurons in the property specific maps \(X_{\text{prop}}\) evolving through equation 23 drives the activity of hub neurons (modulated by the connectivity matrix \(W_{\text{prop}, \text{hub}}\)). At the same time, the sensory layer is the bottom up input to the property specific maps. Since the property specific maps are trained using standard SOM procedure, a Gaussian kernel is used to compare the sensory weights \(s_i\) of neuron \(i\) with current sensor activations \(S\) in order to determine its bottom up activity. So while sensory layer drives property specific neural maps bottom up, the activity of the neurons in the individual neural maps drive the provincial hub bottom up. The third component is the top down component: for the property specific SOM’s the top down input comes from the provincial hub to which they are connected. For the provincial hub the top down component comes from the connector hub (to which it will be linked using exactly the same principles of preferential attachment and temporal coincidence: this will become prominent in later sections hence is just mentioned as “top down” in equation (23). So just like the provincial hub activates the property specific maps, activity in the connector hub can activate the provincial hub (which inversely acts as the bottom up input to the connector hub). Thus, there is always a bidirectional flow of information, as we move upwards information becomes more multimodal and integrated, as we move downwards it becomes more differentiated (to the level of basic properties that are sensed by the sensory layer). The top down input is also biased by a parameter “\(\beta\)” called the bifurcation parameter proposed originally in [29] that plays the role of modulating “how much” of the neural activity in a specific map is governed “top down” and how much “bottom up”. For example, if \(\beta=0\) in equation 23, the system operates only on real sensory input and is not modulated by activity coming from the provincial hub. Recent results from brain imaging (Menon and Bressler, 2010)-
[30] have provided evidence for existence of such dynamic switching between endogenous mental activity and attention driven exogenous activity mediated by anterior insula (AI) and anterior cingulate cortex (ACC). *Computationally the bifurcation parameter has several functions the main being detecting contradictions between one’s anticipations at the current situation and what is actually perceived. In simple terms, if the world does not behave the way we anticipate it should, it may be better to attend to what is happening in the real world and learn new things.*

5.1.3 Bottom Up vs. Top Down neural activations and resulting inferential capabilities

Figure 5.2 (right panel) shows an example of the application of the network dynamics of the “color-word-shape” network in novel situations. The user inputs a sequence of new words “blue cube” (with no such object present in the environment). As we see the activity in the word SOM gradually propagates to the provincial hub and eventually activates the color SOM in a way that was learnt when the robot was presented a “blue container” and shape SOM in a way that was learnt when the robot was presented a “green cube”. But the overall activity in the global system i.e. provincial hub+ color-word-shape maps as a result of the network dynamics triggered by the utterance of a new word “blue cube” resembles what the robot now anticipates that a “blue cube” must be. If a blue cube is really kept in front of the robot, bottom up sensory input and top down anticipation will end up activating the same neurons in every neural map (and resonance between top down and bottom up is enough evidence to confirm that the novel object placed in front of the robot is indeed a “blue cube”). Further any motor behaviour (reaching, grasping, transporting) can be executed also on this novel object (mentioned just by linguistic input by the user).

Figure 5.3 (right panel) presents an interesting scenario where a user now issues a goal to *grasp the “red container”* (both a novel word and at the same time such an object has never been encountered before). The graphs on the top show the temporal evolution of activity in different maps when given a “new word”. The graphs at the bottom show two cases: network activity when a previously unseen object (green container) is kept in front of the robot and when another unseen object “red container” is placed in front of the robot. *In the later case, we can observe that “top down” activity correlates with “bottom up” even though in both cases the object has been never encountered before (and commanded just using linguistic input).* Even if the situation is novel the robot is still able to execute a new user command in the latter case.
(but at the same time in the former case the robot can infer that there is no “red container” placed in front of it, hence quits the goal). Figure 5.3, left panel shows four additional cases that show pattern completion properties of the network. In sum, even in the very basic network consisting of just four neural maps the results demonstrate three aspects:

1) How novel combinations of neural activity can emerge by reconstructing relevant past experiences (relevant means triggered by a partial sensory cue). This perception is also seen as an act of memory and not essentially driven bottom up.;
2) Resonance between “top down” anticipation and “bottom up” sensation leads to inferential mechanisms that can be used to drive goal directed action (here simple cases like reaching, grasping novel objects, unheard words);
3) Contradictions between “top down” and “bottom up” can be used as a stepping stone to learn further and grow the neural maps.
Fig. 5.3: Left panel shows four cases that demonstrate compositionality, modality independence and pattern completion properties depicted by the “color-word-shape-provincial hub” sub network composed of four neural maps. Right panel presents an interesting scenario where the user issues a goal to *reach the “red container”* (both a novel word and at the same time such an object has never been encountered before). The evolving graphs on the top show the temporal evolution of activity in different maps when given a “new word”. The graphs at the bottom show two cases: bottom up network activity (bifurcation parameter =0) when a previously unseen object (green container) is kept in front of the robot and when another unseen object “red container” is placed in front of the robot. *In the later case we can observe that “top down” activity correlates with “bottom up” even though in both cases the object has been never encountered before (and commanded just using linguistic input).*

### 5.1.4 Detecting Contradictions: Switching to attention driven exploration to learn further

A side effect of “top down” and “bottom up” activity being projected on the same neural substrate is the automatic detection of contradictions. This information is crucial and can be used to generate saliency signals to bias attention towards the anomaly and generate exploratory behaviors’ to learn further (to resolve the contradiction). Such mechanisms are important if the robot has to keep learning “cumulatively” and gradually build up its understanding of how the world works. Results from neuroscience [31] provide support for this idea and suggest that anterior insula and anterior cingulate cortex play important role in the saliency detection network of the brain. Perhaps it is already evident when the user issues the goal to grasp the “red container” (figure 5.3 right panel). Comparing the top down and bottom up activity in different neural maps, it is possible to infer that there is a container in the environment but in the first case it is not of the right “color” that was requested by the user,
while in the latter case the goal is realized. Further, the concept system is inherently multimodal. Hence, in addition to mismatch between top down vs. bottom up, contradictions can also occur if information coming from different modalities do not resonate with each other. The proposed computational model also deals with such issues. Figure 5.4 presents some results.

**Fig. 5.4:** Ability to detect and resolve contradictions is built in at every local network. While the results in figure 5.3 show how contradictions can be inferred due to mismatch between ‘top down’ and ‘bottom up’, figure 5.4 presents results where contradictions are caused due to mismatch between information coming from different sensory modalities. Simply, show any infant a potato and say that it is an apple, it should naturally be surprised. The first two examples show similar situations with the robot. When presented with a green sphere along with the word green container inputted by the teacher, there is saliency in the shape, hub and word maps. Note that contradictions are detected locally, in other words the robot infers that there is something green that correlated with the color perceived visually and the word uttered by the teacher, but it also infers that the there is contradictions in the shape and word (it anticipates what it should be associated with the presented object and what the teacher calls it to be). In the second example all maps detect saliency. *In this sense, global saliency of a network is the cumulative sum of local saliencies of individual members.* Greater the global saliency, smaller is the bifurcation parameter greater is the urgency to learn by switching to attention guided exploration.

When presented with a green sphere along with the word green container inputted by the teacher, there is saliency in the shape, hub and word maps. Note that contradictions are
detected locally, in other words the robot infers that there is something green that correlated with the color perceived visually and the word uttered by the teacher, but it also infers that there are contradictions in the shape and word (it anticipates what it should be associated with the presented object and what the teacher calls it to be). In the second example all maps detect saliency. The same applies to the third scenario where an absolutely new object is presented to the robot. As seen from the activity in different neural maps there is no definitive winner (there are multiple hypothesis, hence greater saliency). Saliency can also be thought as a measure of how confused the system is and this applies ‘both’ when there are ‘contradictions’ or when the system is operating in ‘novel situations’. Also note that global saliency of a network is the cumulative sum of local saliencies of individual members. Greater the global saliency greater is the discomfort in the network, greater is the urgency to learn further. The net effect of saliency in terms of the network dynamics is to lower the bifurcation parameter hence causing the switch between endogenous mental simulations to attention driven exogenous exploration. Thus contradictions can be seen as stepping stones to learn new stuff (and the bifurcation parameter drives the switch between endogenous mental simulations to attention driven exogenous exploration). More recently, interesting results are emerging from neuroscience that implicate the fact that the delusional behaviours in neurological disorders (like schizophrenia) are a result of improper mixing of “top down” with “bottom up”. In this background, the bifurcation parameter, now also has a biological basis and significant importance in switching network dynamics between exogenous activity driven by real world and endogenous mental simulations during reasoning about actions, resolving contradictions by either learn more or just reconciling one’s beliefs with what new has been experienced.

5.2 Connecting Objects, Actions and the Body schema

In an embodied framework, “Actions” are mediated through the “Body” and directed towards “Objects” in the environment. Playful interactions with objects give rise to sensorimotor experience, learning and ability to reason. Thus the need to connect “object”, “action” and the body/body schema. The scheme is shown in figure 5.5 and directly builds up on the “object” related “small world” created in section 5.1. Note that there is a subtle separation between representation of actions at an abstract level (“what all can be done with an object/tool”) and the procedural memory related to the action itself (“how to do”). While the former relates to
‘affordance’ of an object, the latter relates to the ‘skill’ of using an object. The abstract layer grasp, push, use of different tools etc, grows with time as new skills are learnt. It consists of single neurons coding for different actions like reach, grasp, etc which they are connected to the “connector hub” and neurons in the connector hub in turn have the capability to trigger the procedural memory network responsible for generating the action they code for. Connector hubs in the object space and action space are connected. All the connections are meant to develop by experience. Connectivity is of the “dual dyad” type hence allowing bidirectional flow of information between different neural maps. As a simple example, consider that an object is presented to the robot with which the robot has some past experience. Then, information from the sensory layer activates various property specific maps, their provincial hubs, finally leading to distributed activity in the object connector hub (which is a multimodal representation of the object: like in figures 5.2-5.3). Assuming that the robot already has experience of performing different actions on this object, the activity in the ‘connector hub’ of the action layer basically codes for what all high level “actions” can be done with this object (more the robot learns more will be the possibilities to exploit an object). In this sense, single neurons in the top level “action connector hub” are similar to “canonical neurons” found in the pre-motor cortex (of monkeys and humans) that are activated at the sight of objects to which specific actions are applicable. At the same time the detailed knowledge itself is learnt/represented in specialized procedural memory networks which are triggered by neurons at the action connector hub. The neural connectivity between top level object and action hub’s are learnt gradually as the robot tries out and learns what actions are possible with an object. This can be due to explorative interaction (for example, a small red cylinder may be reached, grasped, moves in a specific way pushed, etc) or by observing and imitating a teacher like while learning to manoeuvre various tools [32].

Importantly, motor repertoire is gradually built up by interacting with various objects. As a last step, actions have to be ultimately executed by the body and for this we must synthesize the motor commands in the task relevant body chain. This is accomplished by the link between the ‘procedural memory layer’ and ‘body schema’. Action plans (or virtual trajectories) synthesized by the procedural memory networks serve as attractors to the “body schema” hence triggering the PMP simulation in the task relevant body network. As an example, if the task is to rotate a lever, the desired trajectory of motion in the extrinsic space is planned by the procedural
memory network. This acts as a moving point attractor to the task relevant body network of the PMP (for example, the right hand-waist chain). PMP simulation gives out the motor commands which if sent to the actuators produce the desired action. Basic actions like reach, grasp, directed search through vision, use of one tool (a toy crane) to pick up unreachable objects are presently functional with a reasonable level of accuracy. Several primitive actions, reaching (IIT), grasping (KCL), pushing (IIT), visual exploration (CVUT and tool use (IIT and KCL), have been acquired by the robot.
Fig. 5.5: As seen figure 5.5 builds up on figure 5.2 by adding new networks related to “action” and “body schema”. The connectivity and information flows between “object”, “action” and “body” related networks are shown. The information flow is inherently bidirectional and characterized by “dual dyad” type connectivity. There is a subtle separation between representation of actions at an abstract level and the procedural memory network related to the action itself. The abstract layer forms the “connector hub” in the action space and consists of single neurons coding for different actions at an abstract level (like reach, grasp, push, tool use etc). The abstract action layer is similar to “canonical neurons” found in the pre-motor cortex that are activated at the sight of objects to which specific actions are applicable. Note that
these single neurons do not code for the action itself but instead have the capability to trigger the complete procedural memory network responsible for generating the plan to execute the concerned action. All connectivity between various networks is learnt by explorative sensorimotor experience. Specific functions of various layers are summarized in the figure.

In sum, a powerful framework for cumulative learning and organization of concepts related to objects, actions, causal relations has been developed in the DARWIN architecture hence endowing the robot to perceive, act, simulate and infer about the properties of wide range of objects, what can be done with them and how they can be exploited in the context of realization of goals. These issues have been further explored in two recent journal publications [33, 34].
6.0 Conclusions

In this deliverable we have discussed the concept system of the DARWIN architecture. Initially we have presented briefly material related to the experimental findings for concepts as have been captured by numerous psychological experiments assessing performance in various semantic tasks. Section 5 added more evidence from neuroscience. A brief overview of the main psychological theories for concepts was given next. From these findings we summarised the main properties that conceptual representations should respect.

Next we presented a model for extraction of spatial relations between two objects. The purpose of the model is to recursively use it in capturing the whole-part relation of a composite object. The composite object “shape” representation is used as a “feature” of the concept which represents such a composite object. Simulation results were given on objects of year two of DARWIN.

In the next step a representational framework, inspired by work on cognitive science, was put forward in order to support all of the main theories of concepts those of Exemplars, Prototypes and the Theory-theory. Key ideas and discussion on representation issues was given as a founding base where the main computational and information processing model of the framework was developed upon. It is important to note that the framework provides a flexible structure of representation, especially in the level of properties on how concrete representations could be build. Simulation results were given to support the validity of the framework in concrete examples. The development of similar internal representations for concepts showing semantic similarity was discussed and results were presented. The main properties of the representational framework can be summarised as:

- The Concept Model is inspired by and supports all of the main theories of concepts;
- It allows the classification with variable number of “feature” sets;
- It allows for the incremental update of all created concepts;
- It supports forgetting of concepts;
- It builds “summary representations” (prototypes);
- It distinguishes between “typical” cases (prototypes) and “boundary” cases (exemplar);
• Exemplars can “support” many different concepts in parallel;
• Provides “default” values on missing features;
• The model supports properties P1-P4 of the core list of properties of concepts;
• Part of the concept model is a sub-model which captures the relevant semantic information for the concept.
• The aforementioned sub-model is a compressed form of Novak’s “concept maps” which in turn are a model for the Theory-theory view of concepts. This view allows one to recognise concepts which do not have perceptual properties and they are highly abstract.

Based on these ideas a more concrete computational model, inspired by recent findings in neuroscience, was developed for the DARWIN architecture. This model is not only compatible with the representational framework of section 4 but is also biologically plausible. This is a concrete version of the general framework and works well on object representations and through close interaction with the reasoning system it can discover the (apparent) causal relations between perceptual properties and actions leading to movement of objects thus leading to the development of a naïve Theory of Movement.

Even though much progress has been established some open questions remain. For example the definition of a high-level strategy that selects the appropriate values for the specificity and threshold of recognition values given just the data incoming to the system. Ideally this high-level strategy should be domain agnostic and it would automatically select the level of resolution for the overall system, maximizing inference ability while reducing information redundancy at the same time.

A much harder problem is to handle the recognition of functional objects at the same time when concrete ones are present. For example when we present a stack of objects we need to recognise the concept of a “stack” together with the objects that participate to the stack. There is a “moment of transition” when objects, stacked one on top of the other, stop to be just objects (i.e. separate activated object representations with a particular spatial configuration) and they become a “stack”, i.e. a new single object with emergent properties. To the best of our knowledge no perceptual system so far managed to maintain two representations at the
same time: One for stacked objects and one for the resulting “stack” moving from the first to the second, as needed, after the analysis of the visual scene. This is certainly a hard problem as the perceptual system alone could only recognise concrete and not functional objects. Still natural cognitive systems can move from the configuration of stacked objects to the single stack object depending on the context of reasoning and current goals. We have already built the foundation for a possible solution with the introduction of the tree representation for composite objects. We progressed further by discovering in an unsupervised way the stack signature due to its characteristic spatial relation holding true for any two objects in the stack: that of being the one object on top of the other. However, to claim that the system has completely learned the concept in a general setting this means that we should be able to recognise any possible stacks consisting of different objects and of various heights. This indicates rather learning of a “definitional rule” in the level of the Theory-theory representation mode. It is an open question worth investigating further.

Finally the framework presented in section 4 was applied to develop a similar concept model for a commercial application. Even though the system is still under development the inclusion of the Theory-theory representation mode provided much flexibility in comparison to the model in its initial stage of development. The application area is focused in knowledge management and the “perceptual” modalities here include linguistic sub-systems that analyse written language input such as documents, web-pages, etc. In this application domain it is quite difficult to build meaningful content for concepts encountered in texts. One promising way seems to be the concept formation and assimilation strategy suggested by Ausubel. Teaching the system primary concepts that arrive through the experience of a robotic agent can lead possibly to the acquisition of more complex concepts that are assimilated typically through definitional means or through causal extraction of relations. Certainly it is a promising to way to apply DARWIN’s technology to other application domains and in particular in applications of cognitive linguistics.
References


