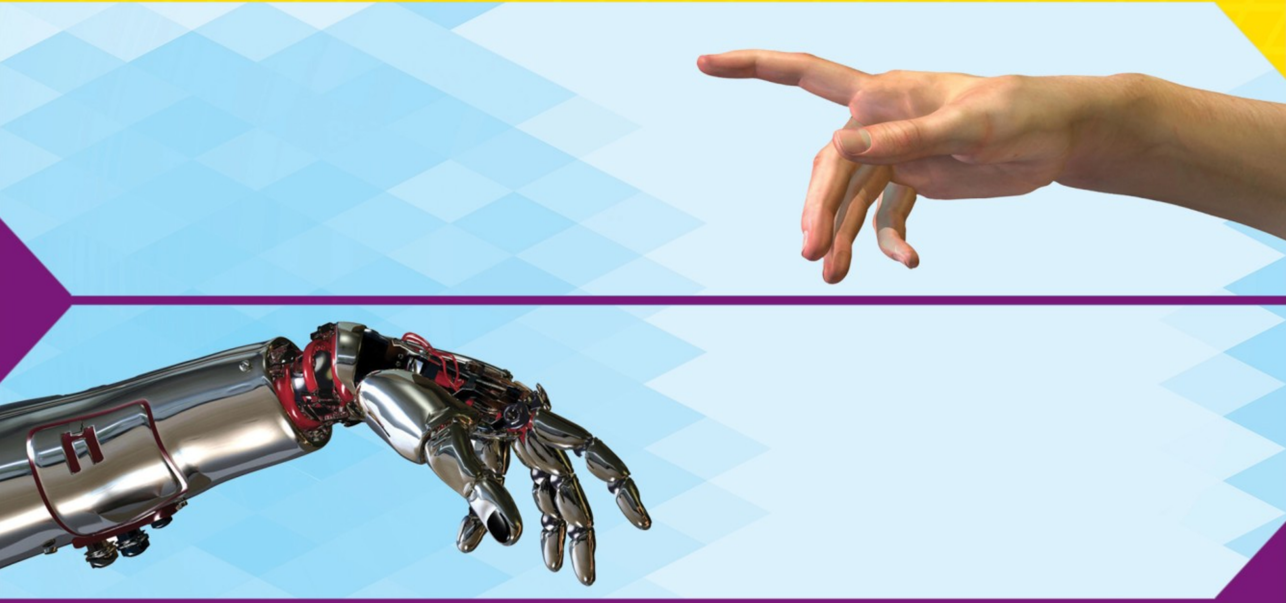

ROBOT LEARNING AND REPRODUCTION OF HIGH-LEVEL BEHAVIORS

BENJAMIN FONOONI

Department of Computing Science
Licentiate Thesis, Umeå 2013

Robot Learning and Reproduction of High-Level Behaviors

Benjamin Fonooni

Licentiate Thesis, September 2013

Department of Computing Science
Umeå University
SE-901 87 Umeå
Sweden

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden

fonooni@cs.umu.se

ISBN: 978-91-7459-712-7
ISSN: 0348-0542
UMINF: 13.20

Electronic version available at http://umu.diva-portal.org/
Printed by: Print & Media, Umeå University
Umeå, Sweden 2013

Abstract

Learning techniques are drawing extensive attention in the robotics community. Some reasons behind moving from traditional preprogrammed robots to learning robots are to save time and energy, and allow non-technical users to easily work with robots. Learning from Demonstration (LfD) and Imitation Learning (IL), in which the robot learns by observing a human or robot tutor, are among the most popular learning techniques.

Flawlessly teaching robots new skills by LfD requires good understanding of all challenges in the field. Studies of imitation learning in humans and animals show that several cognitive abilities are engaged to correctly learn new skills. The most remarkable ones are the ability to direct attention to important aspects of demonstrations, and adapting observed actions to the agents own body. Moreover, a clear understanding of the demonstrator's intentions is essential for correctly and completely replicating the behavior with the same effects on the world. Once learning is accomplished, various stimuli may trigger the cognitive system to execute new skills that have become part of the repertoire.

Considering identified main challenges, the current thesis attempts to model imitation learning in robots, mainly focusing on understanding the tutor's intentions and recognizing what elements of the demonstration need the robot's attention. Thereby, an architecture containing required cognitive functions for learning and reproducing high-level aspects of demonstrations is proposed. Several learning methods for directing the robot's attention and identifying relevant information are introduced. The architecture integrates motor actions with concepts, objects and environmental states to ensure correct reproduction of skills. This is further applied in learning object affordances, behavior arbitration and goal emulation.

The architecture and learning methods are applied and evaluated in several real world scenarios that require clear understanding of goals and what to look for in the demonstrations. Finally, the developed learning methods are compared, and conditions where each of them has better applicability is specified.

Sammanfattning

Tekniker för inlärning uppmärksammas alltmer inom robotikforskningen. Några av fördelarna med robotar som kan lära sig saker, jämfört med traditionella förprogrammerade robotar, är att man sparar tid och kraft, och att icke-tekniska användare enkelt kan arbeta med robotar. Två av de mest populära är Inlärning genom demonstration (Learning from Demonstration – förkortat LfD) och Inlärning genom imitation (Imitation Learning – förkortat IL). Med dessa tekniker lär sig roboten nya färdigheter genom att observera en människa eller robot.

För att utveckla robotar som använder LFD krävs god förståelse av alla utmaningar inom området. Studier av imitationsinlärning hos människor och djur visar att flera kognitiva förmågor är inblandade för korrekt inlärning av nya färdigheter. De mest anmärkningsvärda är förmågan att rikta uppmärksamheten mot de viktiga aspekterna av en demonstration, och att översätta observerade rörelser till robotens speciella fysiska uppbyggnad. Dessutom är en tydlig förståelse av lärarens avsikter viktig för att korrekt och fullständigt kunna replikera ett demonstrerat beteende. När inlärningen är fullbordad, kan olika stimuli utlösa de inlärda färdigheterna.

Med dessa utmaningar i beaktande försöker denna avhandling modellera robotinlärning genom imitation, med fokus främst på att förstå lärarens intentioner och vilka delar av demonstrationen som är viktiga. En arkitektur som innehåller nödvändiga kognitiva funktioner för inlärning och återgivning av högnivåaspekter av demonstrationer presenteras. Flera inlärningsmetoder för att kontrollera robotens uppmärksamhet och identifiera relevant information presenteras. Arkitekturen integrerar motorkommandon med koncept, objekt och tillståndsvariabler för omgivningen. Detta appliceras även på så kallade affordances, behavior arbitration and goal emulation.

Den utvecklade arkitekturen och inlärningsmetoderna används och utvärderas i flera scenarier som kräver att roboten förstår lärarens avsikt, och vad man ska leta efter i demonstrationerna. Slutligen jämförs de utvecklade metoder för inlärning, och de förhållanden under vilka var och en av dem är tillämpliga specificeras.

Preface

This thesis presents techniques and cognitive architectures for Learning from Demonstration (LfD) and Imitation Learning (IL) challenges. High-level learning and reproduction of behaviors is discussed, and our contributions to the field elaborated. The thesis is based on the following papers:

Paper I: Benjamin Fonooni, Thomas Hellström and Lars-Erik Janlert. Learning high-level behaviors from demonstration through Semantic Networks, *In proceedings of 4th International Conference on Agents and Artificial Intelligence (ICAART)*, Vilamoura, Portugal, pp. 419-426, 2012.

Paper II: Benjamin Fonooni, Thomas Hellström and Lars-Erik Janlert. Towards Goal Based Architecture Design for Learning High-Level Representation of Behaviors from Demonstration, *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, San Diego, CA, USA, pp. 67-74, 2013.

Paper III: Benjamin Fonooni, Aleksandar Jevtić, Thomas Hellström and Lars-Erik Janlert. Applying Ant Colony Optimization algorithms for High-Level Behavior Learning and Reproduction from Demonstrations, *to be submitted*, 2013.

In addition to above papers, the following paper has been produced during the PhD studies:

- Benjamin Fonooni. Sequential Learning From Demonstration Based On Semantic Networks, *Umeå's 15th Student Conference in Computing Science (USCCS)*, Umeå, Sweden, 2012.

This work was partly financed by the EU funded Initial Training Network (ITN) in the Marie-Curie People Programme (FP7): INTRO (INTeractive RObotics research network), grant agreement no.: 238486.

Acknowledgments

It would not have been possible to write this thesis without the help and support of many kind people. First, I offer my sincerest gratitude to my supervisor Thomas Hellström who has supported me throughout my PhD career with his inexhaustible patience and knowledge. He is not only a great supervisor who provides invaluable guidance, meticulous suggestions and astute criticism, but a close friend who has always been there for me. Besides Thomas, I would like to thank my co-supervisor Lars-Erik Janlert for his encouragement and insightful comments especially on cognitive aspects of the thesis.

My sincere thanks to Erik Billing for his support and fruitful discussions especially on integrating PSL into my work, all of which were enlightening. I would also like to show my greatest appreciation to Ola Ringdahl for his kindness in answering all my questions.

I wish to take this opportunity to extend my gratitude to everyone at the department of Computing Science for their cooperation, understanding, and support which make it a pleasant working environment. I would also like to express my appreciation to my fellow ESRs and other colleagues from the INTRO for their support and friendship throughout the project. I am particularly grateful for the helpful assistance on Ant Colony Optimization given by Aleksandar Jevtić from Robosoft.

Special thanks to Omid Mohammadian for his splendid job on cover design and putting all parts together to create an outstanding piece of art.

I would also like to express my deepest gratitude to my parents, Behnam and Heideh, for the inspiring education they provided me.

Last but not least, I am grateful to my lovely wife Sepideh, who has been for me all the way with her true love and patience although she went through tough times. Her endeavors to provide peaceful and comforting atmosphere for my studies are undoubtedly affected the current work.

Umeå, August 2013
Benjamin Fonooni

Contents

Chapter 1

Introduction

Robots have already found their way into our lives, and their prospect influence on our daily tasks is irrefutable. Personal robots that can help out with home or office chores are getting popular, and a trend to move away from preprogrammed robots operating in well-defined controlled environment has started. Programming robots for different tasks most often requires considerable cost and energy, and has to be done by experts. Therefore, finding proper solutions based on human's natural ways of learning for efficiently teaching robots new skills can reduce the complexity for end-users as well as saving resources. Humans usually acquire their skills through direct tutelage, observational conditioning, goal emulation, imitation and other social interactions (Scassellati, 1999b). This has opened a new area in human-robot interaction such that even non-roboticist users may teach robots to perform a task by simply showing how to accomplish it with a demonstration. The task can vary from a very simple action of "picking up a cup" to a complex one like "assisting human agent to uncover victim from rubble in debris". The general technique is called Learning from Demonstration (LfD) or Imitation Learning (IL), and has been studied widely over the past decade.

LfD provides a powerful way to speed up learning new skills, as well as blending robotics with psychology and neuroscience to answer cognitive and biological questions, brought to attention by for instance Schaal (Schaal, 1999) and Demiris (Demiris & Hayes, 2002). Despite all its benefits, a number of challenges have to be tackled from different abstraction levels. These challenges and an overview of related works are discussed in chapter 2.

1.1 True imitation

In theory and practice, there are different levels of complexities in imitating behaviors that has been investigated in many studies (Meltzoff, 1988; Miklósi, 1999). A few social learning mechanisms from biological systems have been introduced to extrapolate each complexity. Sometimes these mechanisms are erroneously considered imitation while they are categorized as pseudo-imitation. Such mechanisms are *response facilitation*, *stimulus enhancement*, *goal emulation* and *mimicking* (Fukano et al., 2006).

Response facilitation: A process which observer starts to exhibit a behavior from his existing repertoire by observing others performing the same behavior.

Stimulus enhancement: A mechanism by which an observer starts to exhibit a behavior from his existing repertoire, due to exposure to an object with affordances that draw the observer's attention.

Goal Emulation: A process of witnessing others interacting with an object to achieve certain results without understanding how it is achieved, and then trying to produce the same results with the same object by its own action repertoire.

Mimicking: A mechanism by which an observer starts to copy all actions performed by others without understanding their intentions.

True imitation is gained by reproducing observed actions of others using the same strategy to achieve the same goals. Thus, depending on what type of imitation is concerned, different requirements are needed.

In the current thesis we are interested in understanding intentions of a demonstrator interacting with an object and reproducing the same goals by motor-actions that are hard-coded or learned during observation. Hence, stimulus enhancement and goal emulation are mostly studied.

1.2 Low-level vs. high-level

Imitation learning in robots consists of different abstraction levels that each one refers to one aspect of imitation. Mapping of sensory-motor information that produces an action to be performed by actuators is referred to low-level. In other words, a low-level representation of a learned skill is a set of sensory-motor mappings (Billard et al., 2008). These mapping can produce the same trajectories as observed during demonstrations or might be adapted to robot's morphology but still result in the same actions. A lot of research has addressed the problem of low-level learning and reproduction of behaviors. Among them, (Dillmann, 2004; Ekvall & Kragic, 2005; Calinon et al., 2007; Pastor et al., 2009; Billing & Hellström, 2010; Ijspeert et al., 2013) are especially worth mentioning.

Another aspect of imitation is related to the demonstrator's intentions, goals and objects of attention, which here are considered high-level representations of skills, and sometimes referred to conceptualization or symbolic learning (Billard et al., 2008). Various techniques for learning purpose of demonstration, understanding tutor's intentions and identifying what objects or elements in demonstration are more important have been developed (Mahmoodian et al., 2013; Hajimirsadeghi et al., 2012; Cakmak et al., 2009; Erlhagen et al., 2006; Chao et al., 2011; Jansen & Belpaeme, 2006).

1.3 Objectives

This thesis heads for designing an architecture for learning high-level aspects of demonstrations. Our architecture includes methods to learn tutor's intentions as well as employing techniques to sequentially learn and reproduce motor-skills in order to achieve the same goals. The architecture contains four learning methods coupled with an attentional mechanism to identify the most important elements of the demonstration. These methods are also used to learn object affordances, thereby helping the robot to select appropriate sensory-motor actions in accordance with high-level perceptions. The architecture is then used for behavior arbitration and robot shared control.

1.4 Outline

The remaining chapters are organized as follows: Chapter 2 presents an overview of Imitation learning in robots, challenges and related works. Chapter 3 focuses on cognitive architectures and frameworks proposed in different studies and how they influence the current work. Chapter 4 is about learning methods and how attention mechanism has been developed. Finally, some notes about future works along with summary of contributions are discussed in chapter 5 and 6.

Chapter 2

Learning from Demonstration and Imitation Learning

In order to overcome the challenges in LfD, *"Big Five"* central questions have to be answered: *Who to imitate? When to imitate? How to imitate? What to imitate? How to evaluate a successful imitation?* A thorough investigation of these research questions may lead to construct robots that are able to benefit from utmost potential of imitation learning (Dautenhahn & Nehaniv, 2002). Among these questions *"Who"* and *"When"* to imitate are mostly left unexplored and the majority of approaches are proposed to tackle *"What"* and *"How"* to imitate, which basically refer to learning and encoding skills respectively. In the current thesis we are addressing *"What"* and *"When"* while employing techniques from the *"How"* question.

2.1 Who to imitate

Finding a proper solution for this question requires exhaustive studies in social sciences, since it is strongly connected to the social interactions between an imitator and a demonstrator. Choosing a demonstrator whose behavior can benefit the imitator is essential. Identifying which demonstrator's task is relevant and serves the imitator in some way requires evaluating the performance of the behaviors shown by the selected demonstrator (Alissandrakis et al., 2002).

2.2 When to imitate

This aspect of imitation learning is also tied to social sciences, and is about identifying an appropriate time period to imitate. The imitator has to identify the beginning and end of a shown behavior, as well as deciding if the observed behavior fits in the current context (Alissandrakis et al., 2002).

2.3 What to imitate

Depending on what aspects of a behavior are of interest, different approaches should be applied. In case of *actions*, the demonstrator's movements are relevant, so copying the exact trajectories is important. In other situations, the result and the *effects* of actions are considered important. This means that, the imitator may reproduce the observed behavior with a different set of actions, but the same *goal* is achieved (Zentall, 2001). According to Byrne and Russon (Byrne et al., 1998) there are two different modes of imitation that are distinct from each other: *action level* imitation is about matching minor details and style of sequential acts (i.e. pushing a lever) and *program level* imitation is about copying the structural organization of a complex process (i.e. picking, folding and chewing herbaceous plants shown by apes). The later one requires an ability in imitator to build hierarchical structures in order to learn coordinated sequence of actions to fulfill a goal.

When the robot attempts to imitate, it is crucial to understand which perceptual aspects of the behavior is more relevant. Having the ability to detect saliency and focus on the relevant elements of a demonstrated behavior requires a sophisticated attentional mechanism (Breazeal & Scassellati, 2002b). Different attentional models have been proposed and evaluated. Some models use fixed criteria to selectively direct all computational resources to the elements of the behavior that has the most relevant information (Mataric, 2002), such as a specific color, motion speed or various depth cues (Breazeal & Scassellati, 1999).

In another model that has been used in imitation learning, simultaneous attention to the same object or state in the environment is provided by concept of shared attention (Hoffman et al., 2006; Scassellati, 1999a).

2.4 How to imitate

Once perception is completed and the robot has decided what to imitate, it has to engage an action within its repertoire to exactly replicate the same trajectories or achieve the same results. In case it does not know how to perform the observed action, the robot has to learn it by mapping perceptions into a sequence of motor actions related to its own body. Therefore, embodiment of the robot and its body constraints determine how observed action can be imitated (Alissandrakis et al., 2002). The mismatch between the robot's and the demonstrator's morphology during the mapping process leads to the so called *correspondence problem* (Nehaniv & Dautenhahn, 2002). From a neuroscience perspective, the correspondence problem is explained by *mirror neurons* (Brass & Heyes, 2005; Iacoboni, 2009), which create shared context and understanding of affordances between imitator and demonstrator.

Most robotics research is *a priory* that allows focusing on finding solutions for *"How to imitate"* by constraining design space and thereby fixing what, when and who to imitate (Dautenhahn & Nehaniv, 2002).

2.5 How to evaluate successful imitation

Evaluation of reproduction of a demonstrated behavior determines if the robot was able to correctly answer the questions described above. Sometimes, imitation is considered successful if the correct motor actions have been employed by the robot (Scassellati, 1999b). Most often, evaluation is based on the specific experimental setup and thus it is difficult to make comparisons of different results (Dautenhahn & Nehaniv, 2002). The evaluation might be done by the demonstrator or by an observer with vocal feedback, facial expressions or any other social interactions.

In case of goal oriented imitation, successful imitation is interpreted as achieving the same results by executing correct actions from the observer's repertoire.

2.6 Other challenges

Within the *"Big Five"* questions described above, lie additional challenges that a learning and reproduction system has to provide solutions. These challenges are generalization, learning object affordances and sequence learning that are considered as parts of big five and may or may not addressed separately, but resolving them leads to designing more social and believable robots.

2.6.1 Generalization

An essential feature of any learning system is its ability to generalize. Generalization is a process of observing a set of training examples, identifying the significantly important features common to these examples and forming a concept definition based on these common features (Mitchell et al., 1986). Once a robot has learned to execute a task in a particular situation, should be able to generalize and reproduce the task in different and unseen situations (Calinon & Billard, 2007). In the real world with a dynamic environment, it is crucial to be able to adapt and perform appropriate actions depending on the perceived situation. In contrast to early works in imitation learning that attempted to simply reproduce behaviors that are copies of what have been observed, recent works attempt to generalize across a set of demonstrations.

Generalization can be considered at the sensory-motor level (sometimes referred to as trajectory level), and at the level of sequences of pre-defined motion primitives that accomplishes a task (Billard et al., 2008). In generalization at trajectory level, robot actuator movements are generalized such that the system creates generic representation of the motion for encoding different related movements. The generalization at level of sequences of pre-defined motion primitives is about recognizing task structure in terms of what actions are involved and create generic task structure to execute other related tasks.

For a robot working close to humans in a dynamic environment with several objects and concepts, the capability to generalize one concept to another is essential. This high-level type of generalization is considered in this thesis. For instance, the robot may learn to clean the table when an empty cup is placed on it. The generalization ability helps the robot to perform the cleaning task also when an empty mug is observed on the table. In this way, object affordances are generalized such that even by perceiving objects of different type, the robot correctly performs the correct task.

2.6.2 Sequence learning

Most complex tasks performed by humans comprise sequences of actions executed in the proper order. Therefore, sequence learning plays an important role in human skill acquisition and high-level reasoning (Sun & Giles, 2001). According to Clegg et al when humans learn sequences, the learned information consists of both sequences of stimuli and corresponding sequences of responses (Clegg et al., 1998). Thus, humans react to a stimuli based on the associated learned response. The same principals are considered while developing sequence learning in robots. In robotics, low-level sequence learning of sensory-motor states is done by utilizing Hidden Markov Models (HMM) (Vakanski et al., 2012), Artificial Neural Networks (ANN) (Billard & Hayes, 1999) and Fuzzy Logic (Billing et al., 2012). High-level aspects, such as task goals, are learned by for instance conceptual spaces, which are knowledge representation models for intentions behind demonstrations (Cubek & Ertel, 2012). The Chain Model, a biologically inspired spiking neuron model that aims at reproducing the functionalities of the human mirror neuron system, is proposed by Chersi to encode the final goal of action sequences (Chersi, 2012). In another study, based on reinforcement learning and implicit imitation, sequences of demonstrator's states (e.g. demonstrator's location and limb positions) is used to learn how to combine set of action hierarchies to achieve subgoals and eventually reach the desired goal (Friesen & Rao, 2010). Lee and Demiris (Lee & Demiris, 2011) used stochastic context-free grammars (SCFGs) to represent high-level actions and model human behaviors. First they trained the system with a set of multipurpose low-level actions with HMMs, and then they defined high-level task-independent actions (goals) that comprised previously learned low-level actions as vocabulary. A human-behavior model, with low-level actions associated to symbols, was then created by utilizing SCFG.

In the current thesis, we propose an architecture for goal-based sequence learning and reproduction of high-level representations of behaviors. In our novel approach, semantic relations between observed concepts/objects and executed actions are learned and generalized in order to achieve demonstrated goals (Fonooni et al., 2013a). In Chapter 3, the proposed architecture and related works are discussed.

2.6.3 Learning object affordances

The quality of an object defines its potential for motor actions to be performed on it and obtained upon execution of an action towards the object (Gibson, 1979). Affordances are defined as relations between actions, objects and effects that are used to predict the outcome of an action, plan to reach a goal or to recognize an object and an action. A noteworthy feature of affordances is its dependability on the world and the robot's sensory-motor capabilities. Moreover, it requires a set of primary actions as prior information. In robot imitation learning, affordances have been used for action recognition while interacting with the demonstrator (Montesano et al., 2008). Lopes and colleagues (Lopes et al., 2007) propose a framework for robot imitation based on an affordances model using Bayesian networks to identify the relation between actions, object features and the effects of those actions. Dogar and colleagues (Dogar et al., 2007) developed a goal-directed affordance based framework to allow the robot to observe effects of its primitive behavior on the environment, and create associations between effects, primitive behaviors and environmental situations. The learned associations helped the robot to perform more complex behaviors in the reproduction period. In work by Thomaz and colleagues (Thomaz & Cakmak, 2009), Socially Guided Machine Learning (SGML) was used to investigate the role of the teacher in physical interaction with the robot and the environment in order to learn about objects and what actions or effects they afford. Lee and colleagues (Lee et al., 2009) showed the efficiency of using object affordances in measuring the relevancy of objects to a task and thus helping the robot to engage appropriate low-level action.

In the current thesis we introduce techniques to learn object affordances and employ them to arbitrate a behavior. These techniques are discussed in chapter 4.

Chapter 3

Cognitive Architecture for Robot Imitation Learning

Inside an intelligent system lies a cognitive architecture that defines its infrastructure. In many robotics applications, especially those regarding imitation learning, structures are defined and guidelines for information flow are specified in this architecture. Depending on objectives, hardware design, behavioral repertoire and perceptual inputs, different architectures have been proposed (Breazeal & Scassellati, 2002a; Chella et al., 2006; Gienger et al., 2010; Bandera et al., 2012; Demiris & Khadhouri, 2006). Apart from basic principles of all cognitive architectures, there are common key components in most architectures for robot imitation learning. According to Langley and colleagues (Langley et al., 2009), principles are aspects of an agent which are essential for all mechanisms to work in different application domains: i) short and long-term memories ii) representation of elements residing in these memories iii) functional processes operating on these structures.

Architectures for robot imitation learning contain common key components for cognitive and motor capabilities of the robots. These components are perception, knowledge management, learning and motor command generation. In the following section the above mentioned architectures are discussed briefly.

3.1 Related works

In the study by Breazeal and colleagues (Breazeal & Scassellati, 2002a), several research problems regarding robot imitation learning are outlined. Their generic control architecture was developed for the Cog and Kismet robots. The architecture discriminates low and high-level perceptions based on how much processing requires for the information delivered by each sensor. Learning functionality is not explicitly handled in one specific component but exist in each one of the components. The Attention System is responsible for regulating attention preferences according to motivational states while learning new motor skills. The Behavior System is designed to infer goals and select appropriate behaviors based on perceptions and motivational states. The result of the behavior selection is transferred to the Motor System for execution on the robot. Figure 3.1 depicts the architecture and involved components.

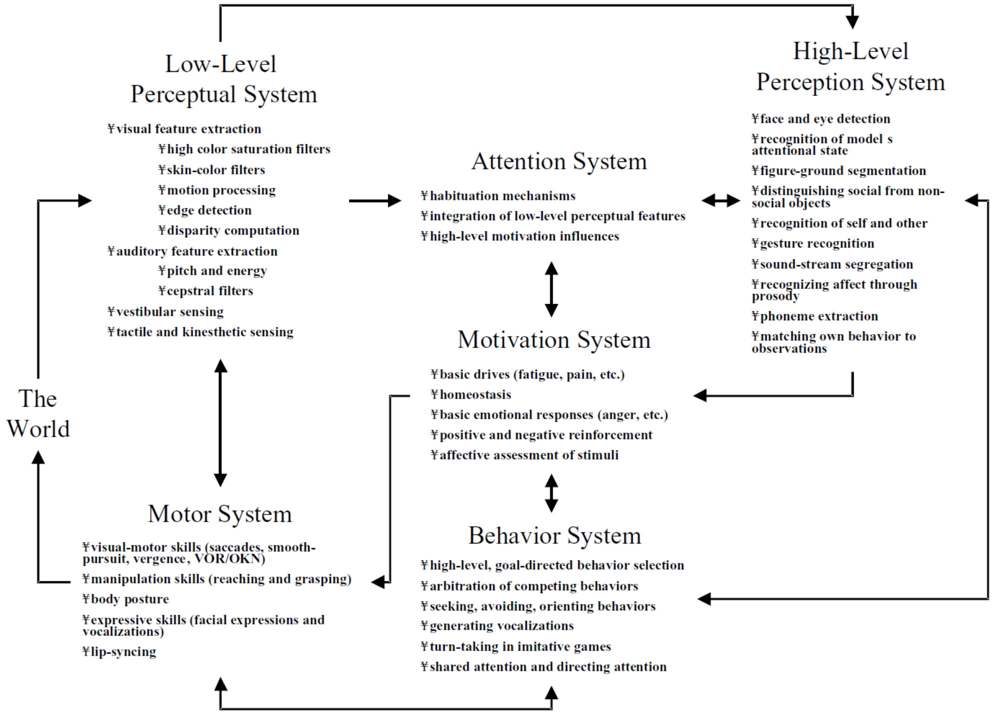

Figure 3.1: Architecture proposed by Breazeal and Scassellati (Breazeal & Scassellati, 2002a) intended to be used on Cog and Kismet.

Chella and colleagues (Chella et al., 2006) proposed an architecture that coupled visual perception with knowledge representation for the purpose of imitation learning. *Conceptual space theory* (Gärdenfors, 2000) is used in their architecture to learn movement primitives from demonstrations and then represent them in generated complex tasks. The architecture functionality has been evaluated on a robotic arm equipped with a camera. Figure 3.2 illustrates the architecture and its components. The architecture consists of three main components. The *Subconceptual Area* is responsible for perception of data from vision sensors, and processing to extract features and controlling robotic system. The *Conceptual Area* is responsible for organizing information provided by the *Subconceptual Area* into categories by using conceptual spaces. Finally, high-level symbolic language has been used to represent sensor data in the *Linguistic Area*. The architecture was designed to work in both observation and imitation modes.

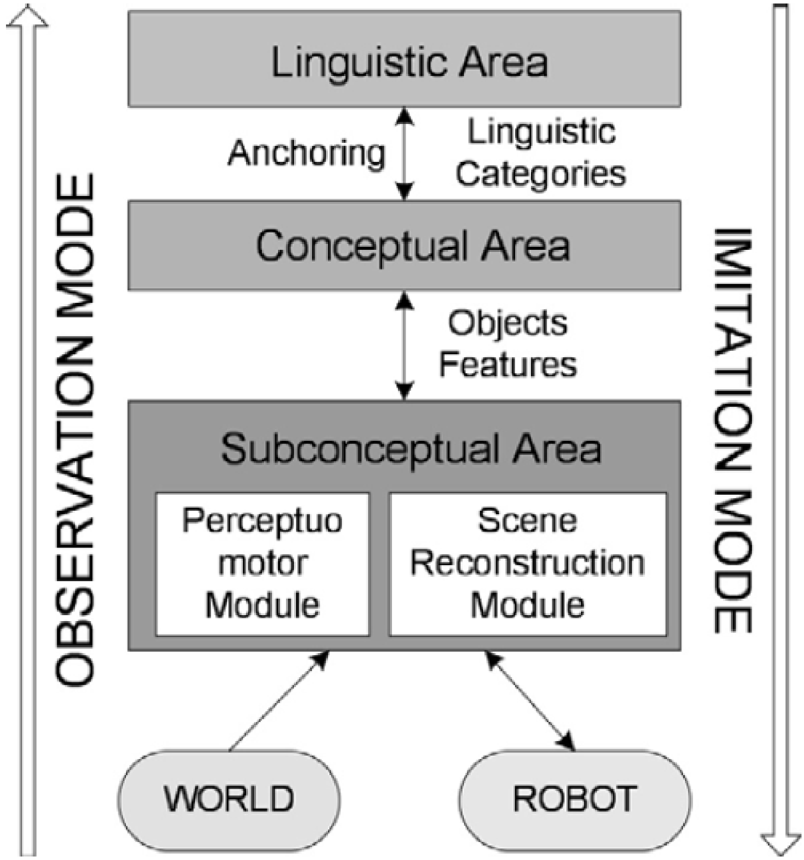

Figure 3.2: Architecture proposed by Chella and colleagues (Chella et al., 2006).

Gienger and colleagues (Gienger et al., 2010) proposed a three-layered architecture based on prior works in the field of imitation learning focusing on movement control and optimization. The aim was to provide solutions for the generalization problem and accomplishing a task in different situations. Figure 3.3 depicts modules that are included within the architecture. The *Reactive* layer is responsible for handling perceptions in the system. *Persistent Object Memory (POM)* was used as an interface between the system and the real world, and includes a model of the world as well as of the robot. While the teacher demonstrates a behavior, the *Movement Primitives* layer normalizes observed movements using *Gaussian Mixture Model (GMM)* and represents them by mean value and variance. Finally, in the Sequence layer, which acts as a procedural memory, sequences of movement primitives are maintained. In the described experiments, predefined primitives for different tasks such as grasping were used, and all learned movements were embedded within predefined locations in the sequence.

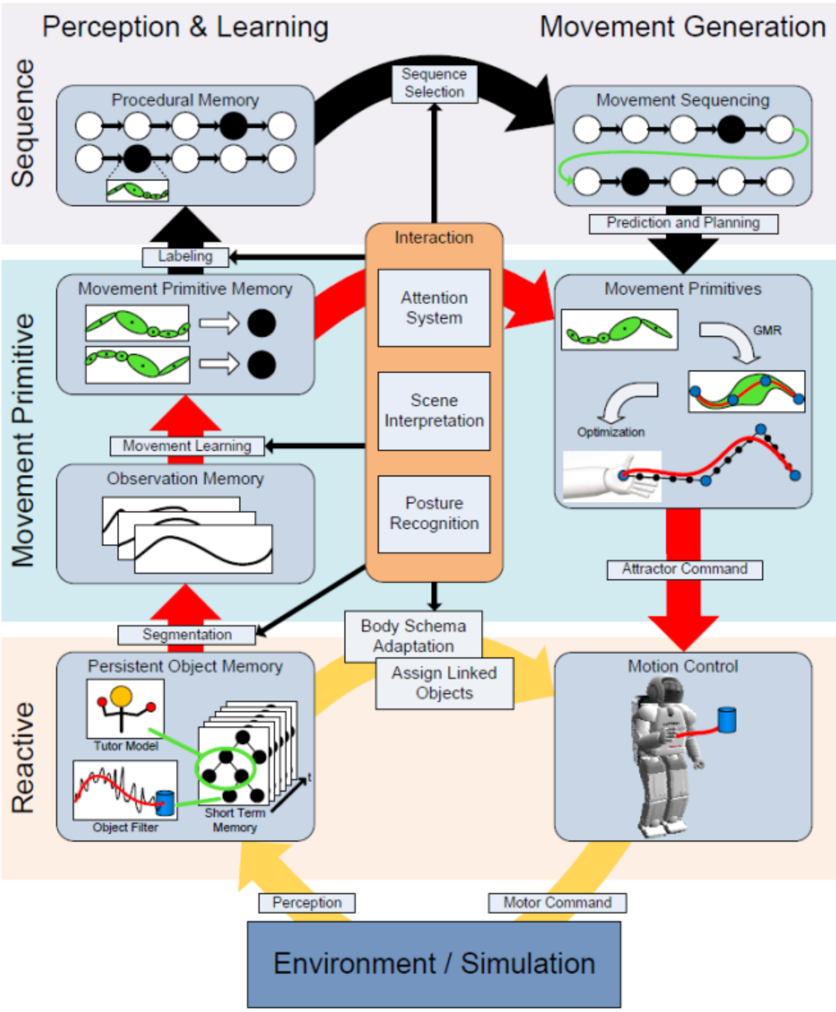

Figure 3.3: Architecture proposed by Gienger and colleagues (Gienger et al., 2010).

In another study by Demiris and Khadhouri (Demiris & Khadhouri, 2006), a hierarchical architecture named HAMMER based on attentive multiple models for action recognition and execution was introduced. As illustrated in Figure 3.4, HAMMER utilizes several inverse and forward models that operate in parallel. Once the robot observes execution of an action, all action states are delivered to the system's available inverse models. Thus, corresponding motor commands representing the hypotheses of which action was demonstrated will be generated and delivered to the related forward model so it can predict the teacher's next movement.

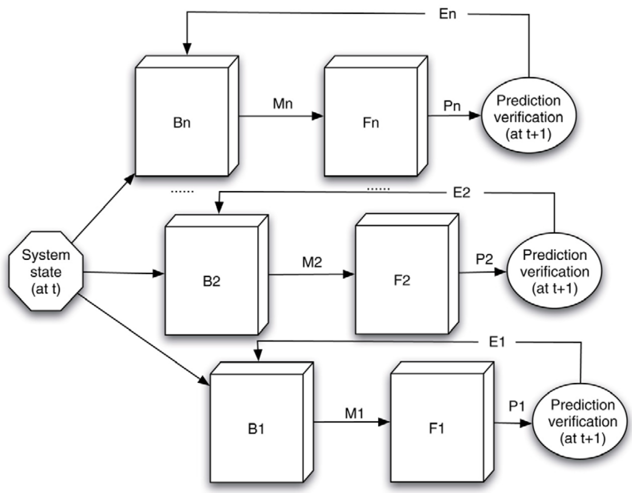

Figure 3.4: The basic architecture proposed by Demiris and Khadhouri (Demiris & Khadhouri, 2006).

Since there might be several possible hypotheses, the attention system is designed to direct the robot's attention to the elements of the action to confirm one of the hypotheses. Figure 3.5 depicts the complete design of the architecture including forward and inverse models together with the attention system for saliency detection. The architecture was tested and evaluated on an ActiveMedia Peoplebot with camera as the only sensor.

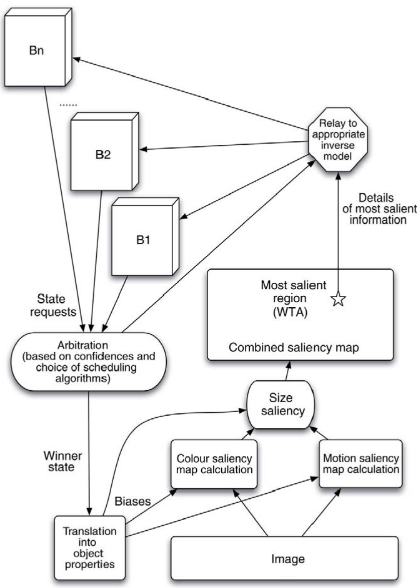

Figure 3.5: The complete architecture proposed by Demiris and Khadhouri (Demiris & Khadhouri, 2006).

In addition to aforementioned studies, other works regarding general cognitive architectures such as ACT-R (Anderson et al., 2004) and SOAR (Laird, 2008), model for reading intentions (Jansen & Belpaeme, 2006) and goal-directed imitation learning frameworks (Tan, 2012) have been reviewed. Furthermore, works by Kopp and Greaser (Kopp & Graeser, 2006) and Buchsbaum and Blumberg (Buchsbaum & Blumberg, 2005) also inspired the design of our novel architecture.

3.2 Proposed Architecture

The rationale behind developing a new architecture while several well-proven ones already exist is a set of new requirements and a new approach to emulating goals in the framework of imitation learning. In the design of our architecture we have considered the hardware setup, robots capabilities and the domain in which they are intended to be used.

Our approach to goal emulation and learning high-level representation of behaviors is to employ a semantic network that contains an ontology of the domain in which the robot is operating, to build semantic relations between robot perceptions and a learned behavior. We named this coupling *context*, and also refer to it as sub-behavior. A context includes presence of objects, concepts and environmental states. During high-level learning, contexts are formed by observing a tutor's demonstration. A complex behavior, also denoted *goal*, consists of several sub-behaviors that are executed in sequence. Not only context formation is taken into consideration during learning but also sequencing. Sequencing is semi-automatic, and comprises one part related to how the tutor conducts the demonstration, and one part related to the system that associates the subsequent context to the preceding one. At the current stage of our architecture development, by finalizing learning of one context and starting learning of another, the system connects both contexts together according to their order in the demonstration.

Once high-level learning is completed, low-level actions will be associated to each one of the learned contexts. Depending on which low-level controller mechanism has been used, the contexts and low-level actions are associated differently. This task is elaborated in section 3.2.3.2. Low-level actions can be learned simultaneously to the contexts, or they can be hard-coded primitives existing in the robot's repertoire. When the complex behavior is reproduced, the actions of each context are executed in the right sequence, initiated by a context selection process.

We have proposed different variations of our architecture, first with low-level learning and control for behavior arbitration (Fonooni et al., 2012) and also with action-primitives and a goal management system to understand the tutor's intentions, as well as behavior arbitration (Fonooni et al., 2013a). Figure 3.6 illustrates the complete architecture and is followed by a description of the individual components.

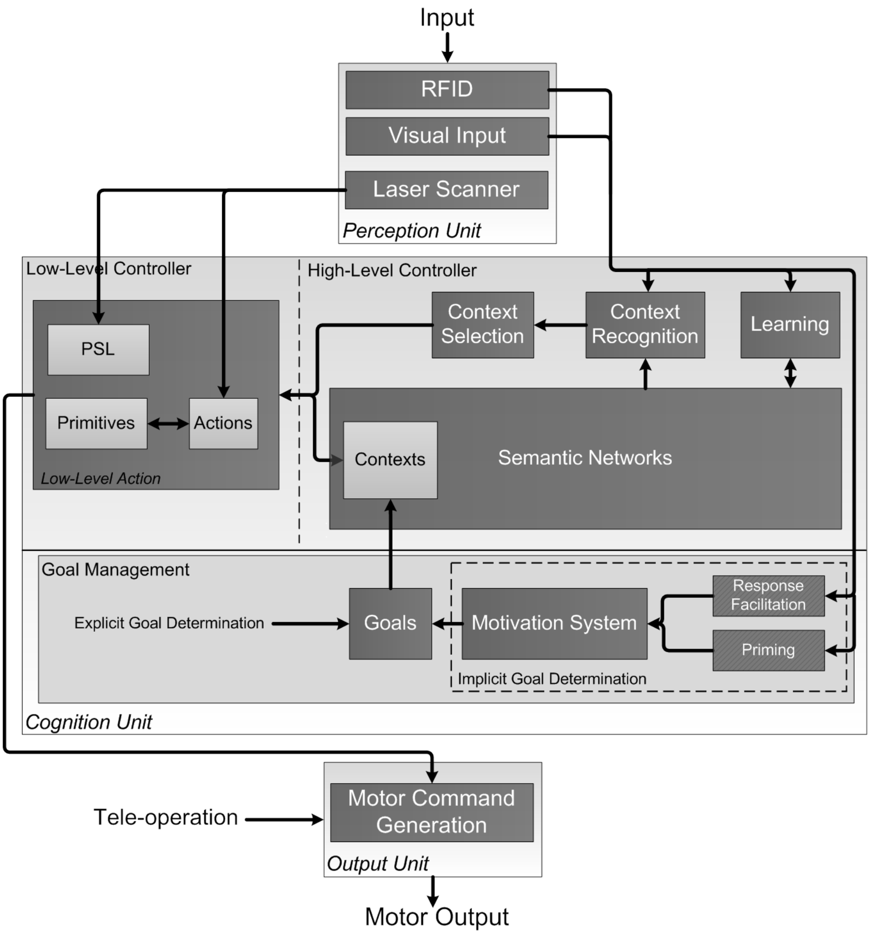

Figure 3.6: The complete architecture developed in the work described in the thesis.

3.2.1 Hardware setup

In all our experiments we used the Robosoft Kompai robot, which is based on the RobuLAB10 platform and robuBOX software (Sallé et al., 2007), as well as Husky A200 Mobile Platform operated by ROS (Quigley et al., 2009). Additional information about our robotic platforms and exhaustive scenario descriptions are well presented in an article written by Jevtić and colleagues from the INTRO project (Jevtic et al., 2012). In order to facilitate the process of object recognition, RFID sensing on the Kompai, and marker recognition tools on the Husky A200 platform were utilized. A database of known objects was linked to the RFID and marker sensors to retrieve properties of the perceived objects. Finally, for mapping and navigation, a laser scanner was used.

3.2.2 Perception unit

All used sensors are included in the perception unit. Sensors are categorized into high and low-level according to the type of information they provide and which controller is the main consumer. Laser data is considered low-level while RFID and marker recognition, included in visual input, are considered high-level. Useful information is extracted from all available input channels by high or low-level controller's request and delivered to the caller in the required format.

3.2.3 Cognition unit

As mentioned earlier, the most common components of all cognitive architectures for imitation learning are knowledge management, learning and control which are also considered in designing of our architecture. The cognition unit is designed such that it can act as the robot's memory for storing both learned and preprogrammed information. It also provides learning facilities with attention mechanisms for recognizing the most relevant cues from perceptions. Making decisions on what actions to perform such that the behavior complies with a specific goal, and providing required structure for behavior arbitration are other tasks for the cognition unit.

3.2.3.1 High-level controller

This module has strong impact on both learning and reproduction of behaviors. Learning a new context, which is an association between the behavior to be learned and perceptions the system regard as relevant, requires an attentional mechanism to identify the most important cues in the demonstrated behavior. Semantic network functions as a long term memory of the robot. The mechanisms for storing and retrieving information from semantic networks are discussed in chapter 4. Each context is part of the semantic network and is represented by a node and semantic relations to all related perceptions represented by links. The *learning* module is connected to the perception unit and also to the semantic network.

Reproduction of a behavior starts by behavior arbitration mechanism which is one of the key aspects of the proposed architecture. By definition, behavior arbitration is a process of taking control from one component of an architecture and delegate it to another (Scheutz, 2002). The robot should reproduce learned behaviors when relevant cues such as environmental states, perceived objects or concepts are present. These cues affect the activation of learned contexts, which control the arbitration process. This is done by *recognizing* all possible contexts that conforms to the assigned goal, and *selecting* the most relevant one to be handed over to low-level controller for action execution. Context learning and the selection processes are thoroughly explained in chapter 4.

3.2.3.2 Low-level controller

This module is responsible for learning and selecting motor actions that are associated to the contexts. In case of learning new action in parallel to learning context, *Predictive Sequence Learning (PSL)* is used. This technique is designed to build a model of a demonstrated sub-behavior from sequences of sensor and motor data during teleoperation, and results in building a hypotheses library. The learned sequences are used to predict which action to expect in the next time step, based on the sequence of passed sensor and motor events during the reproduction phase (Billing et al., 2010). Learning is finalized by associating the learned context with set of hypothesis in the hypotheses library.

In another alternative, learning of motor actions is not considered, and a set of pre-programmed *Action-Primitives* are used. Such a *primitive* is the simplest movement of an actuator in the robot's repertoire that requires a set of parameters for execution. As an example, grasping is a primitive with set of parameters identifying where and how strong to do gripping action with robot's wrist actuator. Depending on the robot's capabilities, different primitives are defined and developed. The *Action* module is an interface between contexts and primitives and retrieves information about the object of attention from the context and passes it as parameters to the primitive in required format. The rationale behind defining actions is the different abstraction levels of contexts and primitives. There are no intersections between the two but they need to be integrated in order to successfully perform a behavior. The main responsibility of the low-level controller during the learning period and using action-primitives, is to identify which primitive has been executed while teleoperating. Thereby, the system can automatically associate the learned context and executed primitive through its action. Every primitive has association to an action which is preprogrammed as well, therefore context is only associated to an action.

In the reproduction phase, once an identified context is delivered from the high-level controller, its corresponding action or hypothesis (depending on whether *Action-Primitives* or *PSL* are used) is identified and passed to the output unit for execution in the robot's actuators.

3.2.3.3 Goal management

This component serves two purposes: i) handling sequences in learning and reproduction of behaviors ii) motivating the robot to reproduce previously learned behaviors by understanding the tutor's intention. As mentioned earlier, throughout the learning process, a complex behavior is decomposed into sub-behaviors, which are demonstrated individually and stored as contexts in the semantic network. Sequence of contexts is also learned while finalizing learning of a sub-behavior and start learning the next one. Therefore, a goal which represents a whole behavior, is created and all contexts in their exact orders are associated to the goal.

Throughout the reproduction phase, a user might explicitly specify a goal for the robot through the designed application user interface. Hence, the robot explores the environment in search for stimuli that activates contexts and thus executes their corresponding actions. The contexts must activate in the same orders as they learned, therefore the robot constantly explores until the required stimulus for activating the right context is perceived. Another form of behavior reproduction is to use the motivation system to implicitly specify a goal for the robot. The motivation system contains *response facilitation* and *priming* mechanisms that put the robot into different tracks. In response facilitation, the robot might initiate a behavior from its repertoire by observing the user exhibiting the same behavior. Therefore, understanding the user's intention and activating the related set of contexts is accomplished through the response facilitation module. Priming is a mechanism that biases the robot to exhibit certain behavior by stimulating the robot with a cue. According to Neely (Neely, 1991) priming is defined as an implicit memory effect that speeds up the response to stimuli because of exposure to a certain event or experience. Anelli and colleagues showed that within the scope of object affordances, priming increases the probability of exhibiting a behavior by observing a related object or concept (Anelli et al., 2012). Once the robot is primed, contexts related to the priming stimuli are activated and, through a bottom-up search from the contexts, the most plausible goal will be identified and selected. Thereby, the actions of the relevant contexts in the selected goal will be engaged in sequence.

3.2.4 Output unit

All actions performed by the robot are executed through the output unit, which retrieves a selected primitive and its set of parameters to generate appropriate motor commands. The ability of robot teleoperation is also critical since this is the way of teaching the robot motor-skills in the proposed architecture.

Chapter 4

Learning high-level representation of behaviors

This chapter presents our learning methods along with attentional mechanism to learn high-level representation of behaviors. The high-level representation of a behavior refers to the aspects of the behavior that consist of goals, intentions and objects of attention. Hence, learning high-level representation of behaviors relates to understanding the tutor's intentions and what elements of the behavior require more attention.

As mentioned earlier, most of the works on high-level learning deal with conceptualization and symbolization. Our approach to conceptualize observed behaviors is to employ Semantic Networks. The robot's perception and understanding of the high-level aspects of behaviors are represented by nodes and their semantic relations. The learning process aims at forming semantic relations of noteworthy concepts, manipulated objects and environmental states throughout the demonstration which we define as context. The role of context is twofold: i) it retains important elements of the learned behavior and thus answers the question of *"what to imitate"* ii) it contains necessary conditions to exhibit a behavior and thus answers the question of *"when to imitate"*. The later one is utilized when the robot perceives the same, or similar, objects or concepts as during learning. This leads to context activation and execution of corresponding actions on the robot.

4.1 Why Semantic Networks

Depending on the field of study, semantics is defined differently. In linguistics it refers to the meaning of words and sentences. In cognitive science it often refers to knowledge of any kind, including linguistic, non-linguistic, objects, events and general facts (Tulving, 1972). Many cognitive abilities like object recognition and categorization, inference and reasoning along with language comprehension are powered by semantic abilities working in semantic memory. Therefore, questions like *"How to understand the purpose of an action?"* or *"How to understand which* *items or events must treated the same?"* without investigating role of semantics ability cannot be answered adequately (Rogers, 2008).

Semantic Networks is a powerful tool to visualize and infer semantic knowledge which is expressed by concepts, their properties and hierarchies of sub and super-class relationships. Semantic Networks have been widely used in many intelligent and robotic systems. In early days hierarchical model of semantic memory was implemented, based on the fact that semantic memory contains variety of simple propositions. An inference engine based on syllogisms was used to deduce new propositional knowledge. Empirical assessment of the proposed model showed that verifying a proposition that is much more common takes more time depending on the number of nodes traversed in the hierarchy (Collins & Quillian, 1969). The typicality was not modeled efficiently in early implementations. For instance, a system could not infer that a chicken is an animal, as fast as it infers that a chicken is a bird. This is due to the hierarchies in the defining semantic relations. But according to Rips and colleagues (Rips et al., 1973), humans are inferring "chicken is an animal" faster due to the typicality that influences the judgment. By revising the early implementations, Collins and Loftus (Collins & Loftus, 1975) introduced a new spreading activation framework that allows direct links from any node to any concept, but with different strengths. This was particularly efficient since it speeded up retrieval of typical information due to their stronger connection, compared to less typical concepts.

4.1.1 Spreading Activation theory

Spreading activation is a process based on a theory of human memory operations that allows propagation of activation from a source node to all its connections according to their strength. Figure 4.1 illustrates the process.

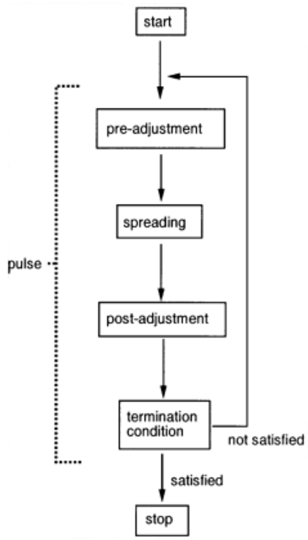

Figure 4.1: The processing technique of spreading activation (Crestani, 1997).

The pre-adjustment and post-adjustment phases are arbitrary since they are both used for activation decay, which may not be applicable in all cases. These phases are responsible for preventing the system to constantly activate certain nodes, and thus implement the concept of *"loss of interest"* (Crestani, 1997). In the spreading phase, the amount of activation to be propagated will be calculated, and all connecting nodes will receive activation according to their strength which is represented by weights.

The pure spreading activation has a few drawbacks, the most remarkable one is the uncontrollable activation propagation that causes the whole network to receive activation (Berthold et al., 2009). To overcome this problem, a system may implement proper pre-adjustment or post-adjustment strategies to avoid spreading activation forever, or may use termination condition to stop spreading at a certain point. But even this is often not sufficient and some other heuristic constraints are commonly used. These constraints are distance, fan-out, path and activation, which also can be used together with termination conditions (Crestani, 1997).

In the current work we use a distance constraint that relates to decreasing activation while spreading in farther levels (distance) from initial node. The rationale behind this constraint is that semantic relations get weaker by distance. We use a decay factor as distance constraint to control how much energy or activation must be subtracted while spreading to each level. For termination condition, an energy value for each node is defined to limit the number of levels the spreading activation process may proceed. This means that spreading continues until a target node not has sufficient energy to continue spreading. The amount of propagated activation is calculated as follows:

$$a_j(t+\Delta t) = \begin{cases} a_j(t) + d\sum w_i a_i(t) & e_i > e_0 \\ \\ a_j(t) & otherwise \end{cases} \tag{4.1}$$

where $a_j(t)$ is activation value of node j at time t,
$a_i(t)$ is activation value of node i, parent of node j, at time t,
Δt is duration of a time step,
$d \in [0,1]$ is decay factor,
and w_i is the weight value of the connection from node i to j and $w_i \in [0,1]$.
The energy level of each node is calculated as follows:

$$e_i(t+\Delta t) = \begin{cases} e_i(t) + d\sum w_n e_n(t) & i \in C_n \\ \\ 0 & otherwise \end{cases} \tag{4.2}$$

where $e_n(t)$ is energy level of parent of node i and $e_i(t) \in [0,1]$,
C_n is a set consist of child nodes of node n.

Since nodes can be connected in loops, firing activation from a node can run forever unless updating of energy values is limited. e_0 denotes an energy threshold that is used to avoid firing activation within a loop of nodes.

4.2 Learning methods

Learning high-level representation of a behavior requires prerequisites including prior knowledge about the domain where the robot is intended to operate. In our case, this knowledge is maintained in a predefined Semantic Network and encompasses many aspects of the domain, such as available objects to manipulate with their respective properties, concepts, environmental states and learned sub-behaviors (contexts). The contexts are also become part of the predefined Semantic Network after learning is complete. Since Semantic Network is used as a model of the world, all items are represented as nodes that have certain properties such as activation values and energy levels that are used for the spreading activation process. Links define semantic relations and contain weight values that are also used in the spreading. Some nodes represent perceivable objects in the environment and are connected to RFID or marker sensors. After each readout, these nodes receive activation and propagate it according to the applied settings. Through the spreading activation mechanism, this results in activation of several nodes, including object features and categories.

The learning process begins with decomposition of the behavior by the tutor into sub-behaviors. Teleoperation is used to demonstrate a sub-behavior to the robot that observes the environment with the sensors. During observation, a learning network is created that contains a new context node connected to all perceived objects and features. Due to the spreading activation process, even non-perceived objects may receive activation and are connected to the context node. All sensors are read within a certain frequency and at each time step, the learning network is updated and activation values of all affected nodes are stored in arrays. In case of demonstrating the same sub-behavior multiple times, learning network and activation arrays of each demonstration are saved separately for further processing. Once all the demonstrations are finished, the system decides on which elements of the demonstrations are most relevant. Since the robot is able to perceive many things that may not be relevant to the goals of the sub-behavior, there is a need for an attentional mechanism to extract important information from the demonstrations. Thereby, we introduced different methods for identifying and removing irrelevant nodes from the final learning network. Based on which method is selected, weight values for remained nodes are calculated. Finally, the predefined Semantic Network is updated according to the remained connections and their corresponding weight values from the learning network. Figure 4.2 depicts all steps in the learning process regardless of which method has been used.

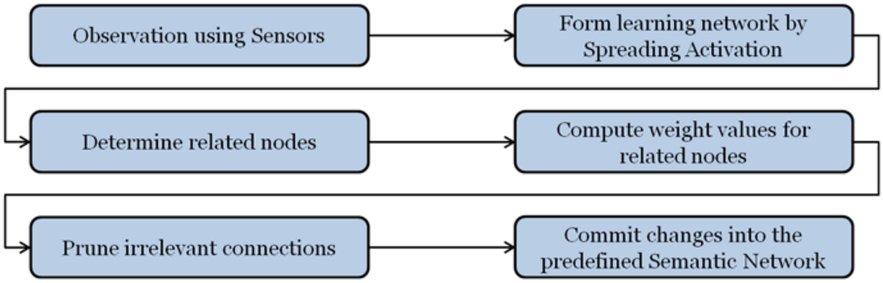

Figure 4.2: Steps of the learning process.

In this thesis, four different context learning methods including mechanisms for directing robot's attention to the relevant elements of demonstrations are introduced.

4.2.1 Hebbian learning

This method is inspired by the well-known Hebbian learning algorithm for artificial neural networks. Its basic tenet is that neurons that fire together, wire together (Hebb, 2002). Hebb suggested that the weight value for the connection between two neurons is proportional to how often they are activated at the same time. In this work, neurons are replaced by nodes in the Semantic Network, and all robot's perceptions are mapped to their corresponding nodes and connected to the context node. This method does not contain any attentional mechanism to identify relevant information but rather keeps all the nodes and strengthen connection of those that are activated together more often.

4.2.2 Novelty Detection

This method is inspired by techniques for detecting novel events in the signal classification domain. While there are many Novelty Detection models available, in practice there is no single best model since it depends heavily on the type of data and statistical features that are handled (Markou & Singh, 2003). Statistical approaches of novelty detection use statistical features to conclude whether data comes from the same distribution or not.

Our approach begins with environment exploration guided by teleoperation to create a history network. In this phase, no demonstrations of desired behaviors are conducted by the tutor, and the history network only contains environmental states. In the next phase, the tutor performs the demonstration and the system builds a learning network accordingly. After collecting required data, a t-test is run to check which nodes have activation values with similar distribution in both history and learning networks. Nodes with different distribution are considered relevant, and thus remain connected to the context node. The weight value of each connection is calculated based on the average node's activation value, and how often the node was received activation during both history and learning phases.

With this approach, the attentional mechanism looks for significant changes between history and learning phases. Nodes that were less, or not at all, activated during the history phase are considered important and most relevant.

In our first paper (Fonooni et al., 2012), we elaborate this technique in detail and evaluate it using a Kompai platform. The test scenario is to teach the robot to push a moveable object to a designated area labeled as storage room.

4.2.3 Multiple Demonstrations

An alternative technique, to some extent the opposite of Novelty Detection is Multiple Demonstrations.The main differences are the number of demonstrations and the way attentional mechanism works. The history phase is removed, and the tutor repeats the demonstration at least two times. In the course of each demonstration, a learning network and activation arrays of nodes are formed and stored. Afterwards, a one-way ANOVA test (Howell, 2011) is run on the datasets of activation values to determine which nodes have different distributions. The attentional mechanism of this method searches for insignificant changes in all demonstrations. Therefore, nodes with least variation in their activations throughout all demonstrations are considered relevant. Weight values are calculated according to nodes average activation values and their presence in all demonstrations.

Paper II (Fonooni et al., 2013a) describes the Multiple Demonstrations technique in an Urban Search And Rescue (USAR) scenario with a Husky A200 platform.

4.2.4 Multiple Demonstrations with ant algorithms

In a variation of the Multiple Demonstrations technique, Ant System (Dorigo et al., 2006) and Ant Colony System (Dorigo & Gambardella, 1997) are used as substitution of the one-way ANOVA test. This technique is showed to be more intuitive and efficient when ANOVA cannot be used to successfully determine the relevant nodes due to statistical constraints. The purpose of applying ant algorithms is to find and strengthen shortest paths that can propagate more activation to the context node. In case of having less intermediate connections (less hierarchies) between the source node that receive activation and the context node, the decay factor has low effect on the amount of propagated activation. Therefore, the closest nodes to the context node are considered more relevant, and thus weight values of remaining connections are calculated based on the amount of laid pheromones.

Paper III (Fonooni et al., 2013b) describes incorporation of Multiple Demonstrations with ant algorithms and presents results from our experiments on learning object shape classification using Kompai robot.

4.2.5 Comparison of methods

Due to the differences between the introduced learning methods, there is no single best method for learning all kinds of behaviors. Therefore, methods have been evaluated according to the type of data they are able to process and scenarios in which they can be more efficient. Table 4.1 lists our learning methods with their respective features and in what conditions they can serve best.

Method	Number of Demonstrations	Core algorithm	Attentional mechanism	Condition
Hebbian Learning	One	Hebbian Learning	None - Nodes that fire together, wire together	When every observation is relevant to the behavior
Novelty Detection	One	Statistical T-test	Looks for significant changes in the history and learning phases	When the robot perceives numerous environmental states that are not relevant to the behavior
Multiple Demonstrations	At least two	One-way ANOVA test	Looks for insignificant changes in all demonstrations	Not noisy environment with only slight differences between demonstrations
Multiple Demonstrations with ACO algorithms	At least two	Ant System (AS) and Ant Colony System (ACS)	Looks for the nodes which can propagate more activation to the context node	Noisy environment where the robot can be easily distracted

Table 4.1: Comparison of learning methods

As Table 4.1 shows, the Hebbian learning approach is used when all perceptions are relevant to the learned sub-behavior. Basically every perception is considered important and must remain connected.

Novelty Detection is mostly successful in situations where the robot is equipped with several sensors and may perceive a large amount of information that is not directly relevant to the behavior. As an example, an ambient light or environment temperature can be sensed if the robot has proper sensors, but this information may not be relevant to the goals of the demonstration. Therefore, the Novelty Detection technique determines what has been static during the history and learning phases and regards these features as unimportant.

Multiple Demonstrations is the best solution if the demonstrations are conducted almost in the same way, and the environment is free from noise. However, if the demonstrations differ significantly, the risk of not recognizing relevant nodes increases dramatically.

Multiple Demonstrations with ant algorithms is more noise tolerant, but still requires that the demonstrations are very similar.

An important limitation with all introduced methods is that none of them are able to learn a behavior that requires understanding of objects absence. Also, quantitative values cannot be handled in a simple way. For instance, learning to clean a table when no human is seated, or to approach a group of people exactly three persons, needs special considerations to be possible with the given techniques.

4.3 Generalization

One of the main challenges in imitation learning is the ability to generalize from observed examples and extend it to novel and unseen situations. Generalization in this work refers to extending associations of objects and concepts that already connected to the context node to less-specific ones. Figure 4.3 shows an example of generalization in terms of extending concepts for learning to find human.

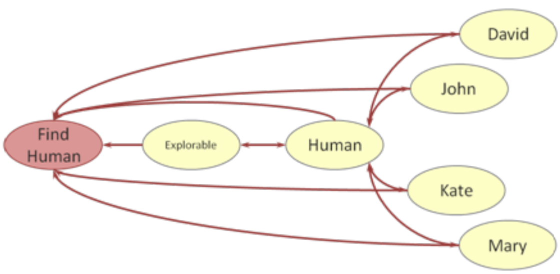

Figure 4.3: Concept generalization example.

In the given example, the robot learns to look for a human and stop exploration when it reaches "John". This will associate the "John" node to the "Find Human" context node. The system correctly associates perceptions to the context, but what the robot learned cannot be used in any other situation. Therefore, generalization of the "John" concept is needed if the intention is to teach the robot to repeat the behavior when any humans observed. Generalization is achieved by spreading activation from "John" node to the less-specific "Human" node. As a result of spreading activation, "Human" is also considered part of the context and thus observing any humans will trigger the "Find Human" context. According to equations (1) and (2), the degree of generalization is controlled by each node's energy value and decay factor. Setting the decay factor to 1.0 leads to generalization to the entire network, and setting it to 0.0 results in no generalization.

Chapter 5

Future Works

Based on the current achievements on designing and implementing an architecture for robot's imitation learning including methods for high-level learning and control, several directions are considered as future works:

- *Extension of current learning methods:* As stated earlier there are known limitations to our learning methods that make them inefficient under certain circumstances. To overcome the issues, all abilities of Semantic Networks must be employed. This includes implementing new type of links such as inhibitory to define negations.

- *Designing new learning methods:* Although spreading activation along with energy values and decay factor are showed to be efficient in our learning strategy, designing a new method based on Fuzzy Logic is also of interest. In this method, membership values assigned to each node determine their relevancy to different contexts. Therefore, all nodes are connected to all contexts but with different membership values.

- *Dealing with ambiguity in demonstrations:* From the outset and throughout this thesis we have assumed that the tutor demonstrates behaviors completely and correctly. However, in reality there are major issues that restraint the robot from perfect learning. One important issue is *ambiguity*, which in the robotics community has several different meanings. Often it is related to insufficient sensing or perception, such that one demonstration maps to several possible behaviors. Differences in robot and teacher perspectives during demonstration may lead to ambiguity due to visual occlusion (Breazeal et al., 2006). Multiple, inconsistent, demonstrations is another cause for ambiguity (Argall et al., 2009). We will investigate ambiguity that occurs when a demonstration contains irrelevant information such that the intention of the tutor is not uniquely described by a single demonstration (Bensch & Hellström, 2010). Since we are mostly interested in high-level representations of behaviors, a solution can be implemented in the developed cognitive architecture. The priming mechanism for implicitly specifying goals during the reproduction phase can act as a bias in the identification of the tutor intentions during the learning phase. The robot may be primed with objects, features, or concepts that directly or indirectly relate to the main objectives of the demonstration. In this way, the attention is directed towards elements that are relevant for the learning. This makes it possible to recognize the tutor's intentions in a less ambiguous way.

- *Robot shared control and imitation learning:* One way to make a robot perform a learned or pre-programmed behavior is to let it observe a user starting to demonstrate the behavior. The robot then attempts to predict the user's next actions based on its repertoire of behaviors. Depending on how successful the predictions are, the robot may then take over control from the user. This gives the user more freedom to engage with other tasks. The user may at any time take over control of the robot.

Chapter 6

Contributions

As stated earlier, the main contribution of this thesis is the design of an architecture for robot's imitation learning that specializes in learning and reproduction of high-level representations of behaviors. The architecture involves learning methods with attention mechanisms based on Semantic Networks and spreading activation theory for identifying important elements of demonstrations as well as recognizing the tutor's intentions. Furthermore, integration of low and high-level learning, techniques for sequence learning and reproduction of skills considering the tutor's goals are described. This infrastructure has been employed for the purpose of behavior arbitration.

The included papers mainly focus on elaborating learning methods and architecture design plus their applicability in real scenarios.

6.1 Paper I

In this paper (Fonooni et al., 2012), a rudimentary architecture for learning high-level representation of behaviors is introduced. The aim is to integrate *Predictive Sequence Learning (PSL)* as low-level learning and control mechanism with a high-level controller that focuses on replicating demonstrated tasks with no knowledge about goals or intentions. The Novelty Detection technique with attentional mechanism based on semantic networks, spreading activation and statistical t-test is introduced.

The system is tested and evaluated with a scenario, in which the goal is to push a movable object towards a designated area. While the task is demonstrated with a certain object, with its particular features, the system is able to generalize, and reproduce the task by observing different, but similar, types of objects. Thereby, the proposed architecture and Novelty Detection showed to be suitable for the purpose of learning and generalizing object affordances. Reproduction of learned behaviors is engaged by exploring the environment and observing any related object. Therefore, stimulus enhancement is applied as a mechanism to trigger behaviors in the robot's repertoire.

6.2 Paper II

In the second paper (Fonooni et al., 2013a), general challenges of LfD at both low and high levels are investigated. Several improvements are made to the architecture, with the aim of facilitating intention recognition and goal emulation. In the developed architecture, PSL is replaced by hard-coded action-primitive pairs that do not require learning. The Multiple Demonstrations technique that uses one-way ANOVA test and spreading activation theory is presented. The attention mechanism with the same impact on learning behaviors as presented in the first paper, is applied with the slight changes in the way relevant information is detected in demonstrations. The goal management module with goal creation and inference capabilities is added. Motivating the robot to exhibit a previously learned behavior with priming mechanism is elaborated. The whole architecture showed to be efficient for sequence learning and reproduction, by decomposing sequences into sub-behaviors that are associated to action-primitive pairs.

Finally, an Urban Search and Rescue (USAR) scenario is defined to evaluate the applicability of the proposed architecture and the learning method. The goal is to assist a human agent to uncover a victim from a pile of rubble in an environment damaged by an earthquake. Results show the system's ability in learning and reproduction of such complex tasks.

6.3 Paper III

The third paper (Fonooni et al., 2013b), attempts to answer the question *"What to imitate?"*, by extending the Multiple Demonstrations technique introduced in the second paper. This technique has practical limitations that prevent the robot to correctly determine what elements of demonstration are mostly important. In this paper, this is viewed as an optimization problem. The one-way ANOVA test is replaced by ant colony optimization algorithms and thus Ant System (AS) and Ant Colony System (ACS) have been utilized. The main contribution of the paper is to investigate the applicability of AS and ACS for identification of relevant nodes, and thereby optimizing the context formation process. Moreover, generalization of concepts by means of spreading activation and ant colony optimization algorithms is investigated.

Although low-level learning and control is not directly addressed, the proposed method can be applied with both PSL and action-primitive pairs.

The whole learning and reproduction mechanisms is tested in a scenario in which the robot learns to identify cylindrical, square and triangular shapes and put them in their respective baskets. Results show that both the AS and ACS algorithms prove to be powerful alternatives to the previously developed Multiple Demonstrations techniques combined with one-way ANOVA test.

Bibliography

Alissandrakis, A., Nehaniv, C. L., & Dautenhahn, K. (2002). Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 32(4), 482–496.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological review*, 111(4), 1036–1060.

Anelli, F., Nicoletti, R., Kalkan, S., Sahin, E., & Borghi, A. M. (2012). Human and robotics hands grasping danger. In *Neural Networks (IJCNN), The 2012 International Joint Conference on* (pp. 1–8).: IEEE.

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469–483.

Bandera, J., Rodríguez, J., Molina-Tanco, L., & Bandera, A. (2012). A survey of vision-based architectures for robot learning by imitation. *International Journal of Humanoid Robotics*, 9(1).

Bensch, S. & Hellström, T. (2010). On ambiguity in robot learning from demonstration. (pp. 47–56).

Berthold, M. R., Brandes, U., Kötter, T., Mader, M., Nagel, U., & Thiel, K. (2009). Pure spreading activation is pointless. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 1915–1918).: ACM.

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Handbook of robotics chapter 59: Robot programming by demonstration.

Billard, A. & Hayes, G. (1999). Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behavior*, 7(1), 35–63.

Billing, E., Hellström, T., & Janlert, L.-E. (2012). Robot learning from demonstration using predictive sequence learning. *Intech Robotic Systems-Applications, Control and Programming, Dr. Ashish Dutta (Ed.).*

Billing, E. A. & Hellström, T. (2010). A formalism for learning from demonstration. *Paladyn Journal of Behavioral Robotics*, 1(1), 1–13.

Billing, E. A., Hellstroöm, T., & Janlert, L.-E. (2010). Behavior recognition for learning from demonstration. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (pp. 866–872).: IEEE.

Brass, M. & Heyes, C. (2005). Imitation: is cognitive neuroscience solving the correspondence problem? *Trends in cognitive sciences*, 9(10), 489–495.

Breazeal, C., Berlin, M., Brooks, A., Gray, J., & Thomaz, A. L. (2006). Using perspective taking to learn from ambiguous demonstrations. *Robotics and Autonomous Systems*, 54(5), 385–393.

Breazeal, C. & Scassellati, B. (1999). A context-dependent attention system for a social robot. (pp. 1146–1151).

Breazeal, C. & Scassellati, B. (2002a). Challenges in building robots that imitate people. *Imitation in animals and artifacts*, (pp. 363–390).

Breazeal, C. & Scassellati, B. (2002b). Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11), 481–487.

Buchsbaum, D. & Blumberg, B. (2005). Imitation as a first step to social learning in synthetic characters: a graph-based approach. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (pp. 9–18).: ACM.

Byrne, R. W., Russon, A. E., et al. (1998). Learning by imitation: a hierarchical approach. *Behavioral and brain sciences*, 21(5), 667–684.

Cakmak, M., DePalma, N., Arriaga, R., & Thomaz, A. L. (2009). Computational benefits of social learning mechanisms: Stimulus enhancement and emulation. In *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on* (pp. 1–7).: IEEE.

Calinon, S. & Billard, A. (2007). Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction* (pp. 255–262).: ACM.

Calinon, S., Guenter, F., & Billard, A. (2007). On learning, representing, and generalizing a task in a humanoid robot. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(2), 286–298.

Chao, C., Cakmak, M., & Thomaz, A. L. (2011). Towards grounding concepts for transfer in goal learning from demonstration. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2 (pp. 1–6).: IEEE.

Chella, A., Dindo, H., & Infantino, I. (2006). A cognitive framework for imitation learning. *Robotics and Autonomous Systems*, 54(5), 403–408.

Chersi, F. (2012). Learning through imitation: a biological approach to robotics. *Transactions on Autonomous Mental Development*, 4(3), 204–214.

Clegg, B. A., DiGirolamo, G. J., & Keele, S. W. (1998). Sequence learning. *Trends in Cognitive Sciences*, 2(8), 275–281.

Collins, A. M. & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological review*, 82(6), 407–428.

Collins, A. M. & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8(2), 240–247.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6), 453–482.

Cubek, R. & Ertel, W. (2012). Conceptual similarity as a key to high-level robot programming by demonstration. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on* (pp. 1–6).: VDE.

Dautenhahn, K. & Nehaniv, C. L. (2002). *The agent-based perspective on imitation.* MIT Press.

Demiris, J. & Hayes, G. (2002). Imitation as a dual-route process featuring predictive and learning components; biologically plausible computational model. *Imitation in animals and artifacts*, (pp. 327–361).

Demiris, Y. & Khadhouri, B. (2006). Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and autonomous systems*, 54(5), 361–369.

Dillmann, R. (2004). Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2), 109–116.

Dogar, M. R., Cakmak, M., Ugur, E., & Sahin, E. (2007). From primitive behaviors to goal-directed behavior using affordances. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (pp. 729–734).: IEEE.

Dorigo, M., Birattari, M., & Stützle, T. (2006). Ant colony optimization - artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, 1(4), 28–39.

Dorigo, M. & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1), 53–66.

Ekvall, S. & Kragic, D. (2005). Grasp recognition for programming by demonstration. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (pp. 748–753).: IEEE.

Erlhagen, W., Mukovskiy, A., Bicho, E., Panin, G., Kiss, C., Knoll, A., Van Schie, H., & Bekkering, H. (2006). Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning. *Robotics and autonomous systems*, 54(5), 353–360.

Fonooni, B., Hellström, T., & Janlert, L.-E. (2012). Learning high-level behaviors from demonstration through semantic networks. In *Proceedings of 4th International Conference on Agents and Artificial Intelligence* (pp. 419–426).

Fonooni, B., Hellström, T., & Janlert, L.-E. (2013a). Towards goal based architecture design for learning high-level representation of behaviors from demonstration. In *2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)* (pp. 67–74).

Fonooni, B., Jevtic, A., Hellström, T., & Janlert, L.-E. (2013b). Applying ant colony optimization algorithms for high-level behavior learning and reproduction from demonstrations.

Friesen, A. L. & Rao, R. P. (2010). Imitation learning with hierarchical actions. In *Development and Learning (ICDL), 2010 IEEE 9th International Conference on* (pp. 263–268).: IEEE.

Fukano, R., Kuniyoshi, Y., & Nagakubo, A. (2006). A cognitive architecture for flexible imitative interaction using tools and objects. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on* (pp. 376–381).: IEEE.

Gibson, J. J. (1979). *The ecological approach to visual perception.* Houghton Mifflin.

Gienger, M., Muhlig, M., & Steil, J. J. (2010). Imitating object movement skills with robots - a task-level approach exploiting generalization and invariance. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on* (pp. 1262–1269).: IEEE.

Gärdenfors, P. (2000). *Conceptual Spaces.* MIT Press.

Hajimirsadeghi, H., Ahmadabadi, M. N., Araabi, B. N., & Moradi, H. (2012). Conceptual imitation learning in a human-robot interaction paradigm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2), 28.

Hebb, D. O. (2002). *The organization of behavior: A neuropsychological theory.* Psychology Press.

Hoffman, M. W., Grimes, D. B., Shon, A. P., & Rao, R. P. (2006). A probabilistic model of gaze imitation and shared attention. *Neural Networks*, 19(3), 299–310.

Howell, D. C. (2011). *Statistical methods for psychology.* Cengage Learning.

Iacoboni, M. (2009). Imitation, empathy, and mirror neurons. *Annual review of psychology*, 60, 653–670.

Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., & Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2), 328–373.

Jansen, B. & Belpaeme, T. (2006). A computational model of intention reading in imitation. *Robotics and Autonomous Systems*, 54(5), 394–402.

Jevtic, A., Lucet, E., Kozlov, A., & Gancet, J. (2012). Intro: A multidisciplinary approach to intelligent human-robot interaction. In *World Automation Congress (WAC), 2012* (pp. 1–6).: IEEE.

Kopp, S. & Graeser, O. (2006). Imitation learning and response facilitation in embodied agents. In *Intelligent Virtual Agents* (pp. 28–41).: Springer.

Laird, J. E. (2008). Extending the soar cognitive architecture. *Frontiers in Artificial Intelligence and Applications*, 171, 224–235.

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, 10(2), 141–160.

Lee, K. & Demiris, Y. (2011). Towards incremental learning of task-dependent action sequences using probabilistic parsing. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2 (pp. 1–6).: IEEE.

Lee, K. H., Lee, J., Thomaz, A. L., & Bobick, A. F. (2009). Effective robot task learning by focusing on task-relevant objects. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (pp. 2551–2556).: IEEE.

Lopes, M., Melo, F. S., & Montesano, L. (2007). Affordance-based imitation learning in robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (pp. 1015–1021).: IEEE.

Mahmoodian, M., Moradi, H., Ahmadabadi, M. N., & Araabi, B. N. (2013). Hierarchical concept learning based on functional similarity of actions. In *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on* (pp. 1–6).: IEEE.

Markou, M. & Singh, S. (2003). Novelty detection: a review - part 1: statistical approaches. *Signal processing*, 83(12), 2481–2497.

Mataric, M. J. (2002). Sensoryf: Linking perception to action and biology to robotics. In *Imitation in Animals and Artifacts* (pp. 391–422).: MIT Press.

Meltzoff, A. N. (1988). Imitation, objects, tools, and the rudiments of language in human ontogeny. *Human evolution*, 3(1-2), 45–64.

Miklósi, A. (1999). The ethological analysis of imitation. *Biological Reviews*, 74(3), 347–374.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine learning*, 1(1), 47–80.

Montesano, L., Lopes, M., Bernardino, A., & Santos-Victor, J. (2008). Learning object affordances: from sensory–motor coordination to imitation. *Robotics, IEEE Transactions on*, 24(1), 15–26.

Neely, J. H. (1991). Semantic priming effects in visual word recognition: A selective review of current findings and theories. *Basic processes in reading: Visual word recognition*, 11, 264–336.

Nehaniv, C. L. & Dautenhahn, K. (2002). The correspondence problem. *Imitation in animals and artifacts*, (pp. 41–61).

Pastor, P., Hoffmann, H., Asfour, T., & Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on* (pp. 763–768).: IEEE.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3.

Rips, L. J., Shoben, E. J., & Smith, E. E. (1973). Semantic distance and the verification of semantic relations. *Journal of verbal learning and verbal behavior*, 12(1), 1–20.

Rogers, T. (2008). Computational models of semantic memory. *The Cambridge handbook of computational psychology*, (pp. 226–266).

Sallé, D., Traonmilin, M., Canou, J., & Dupourqué, V. (2007). Using microsoft robotics studio for the design of generic robotics controllers: the robubox software. In *ICRA 2007 Workshop Software Development and Integration in Robotics - "Understanding Robot Software Architectures", Roma, Italy.*

Scassellati, B. (1999a). Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot. In *Computation for metaphors, analogy, and agents* (pp. 176–195). Springer.

Scassellati, B. (1999b). Knowing what to imitate and knowing when you succeed. In *Proceedings of the AISBŠ99 Symposium on Imitation in Animals and Artifacts* (pp. 105–113).

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6), 233–242.

Scheutz, M. (2002). Affective action selection and behavior arbitration for autonomous robots. In *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, volume 1 (pp. 334–340).

Sun, R. & Giles, C. L. (2001). Sequence learning: From recognition and prediction to sequential decision making. *IEEE Intelligent Systems*, 16(4), 67–70.

Tan, H. (2012). Implementation of a framework for imitation learning on a humanoid robot using a cognitive architecture. *The Future of Humanoid Robots: Research and Applications*, (pp. 189–210).

Thomaz, A. L. & Cakmak, M. (2009). Learning about objects with human teachers. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction* (pp. 15–22).: ACM.

Tulving, E. (1972). Episodic and semantic memory. *Organization of memory*, (pp. 381–402).

Vakanski, A., Mantegh, I., Irish, A., & Janabi-Sharifi, F. (2012). Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(4), 1039–1052.

Zentall, T. R. (2001). Imitation in animals: evidence, function, and mechanisms. *Cybernetics & Systems*, 32(1-2), 53–96.

I

LEARNING HIGH-LEVEL BEHAVIORS FROM DEMONSTRATION THROUGH SEMANTIC NETWORKS

Benjamin Fonooni[1], Thomas Hellström[1] and Lars-Erik Janlert[1]
[1] *Department of Computing Science, Umeå University, SE 901 87, Sweden*
{fonooni, thomash, lej}@cs.umu.se

Keywords: Learning from Demonstration, High-Level Behaviors, Semantic Networks, Robot Learning

Abstract: In this paper we present an approach for high-level behavior recognition and selection integrated with a low-level controller to help the robot to learn new skills from demonstrations. By means of Semantic Network as the core of the method, the robot gains the ability to model the world with concepts and relate them to low-level sensory-motor states. We also show how the generalization ability of Semantic Networks can be used to extend learned skills to new situations.

1 INTRODUCTION

Learning from Demonstration (LfD) is a technique to teach robots new behaviors by having a human or robot teacher performing sequences of actions that are either observed or perceived by the robot. Several algorithms have been proposed. Most of them distinguish between low and high-level representations of a behavior (see for instance Billard et al., 2008). In our approach, the low-level is represented by sensory-motor mappings and the high-level by combinations of concepts represented in Semantic Networks.

One of the hard problems in LfD is how to generalize a demonstrated behavior such that it can be performed also in new, previously unseen situations. This issue exists from both high and low-level perspectives and there are several ways to approach it (Nehaniv and Dautenhahn, 2000; Byrne and Russon, 1998). The purpose of this paper is to introduce a technique that integrates high and low-level learning and control in a way that supports generalization. A high-level controller deals with concepts represented and processed in Semantic Networks (SN). This controller is interfaced to a low-level controller that learns and performs behaviors defined at the sensory-motor level. The glue, interfacing the two levels, is learned *contexts*, describing the necessary high-level conditions for a low-level behavior to be performed.

Behavioral studies of animals and humans provide several sources for ideas on how low and high-level learning may be combined. For instance, *Goal emulation* (Whiten and Ham, 1992) is interesting for learning how to direct focus of attention towards favorable goals. *Stimulus enhancement* is the implicit memory effect when stimuli in the environment bias the agent's behavior towards receiving similar stimuli in the future. *Response facilitation* is a mechanism that describes how motor responses already in the repertoire be repeated after observing the performance of the same action (Kopp and Graeser, 2006). In a broad sense, the work presented in this paper may be seen as a realization of response facilitation. All mechanisms described above may be seen as examples of priming, aiming at guiding animal behavior and learning (Byrne, 1994).

In section 2, the proposed architecture with its major units is described. Section 3 is an overview on Semantic Networks and its features. Section 4 elaborates the learning and performing phases based on proposed architecture and Semantic Networks. In section 5, an example is shown to evaluate learning and performing phases.

2 OVERVIEW OF THE ARCHITECTURE

A number of architectures and frameworks for LfD have been developed during passed years which influenced the current research in this field (Kasper et al., 2001; Nicolescu, 2003). These works are introducing architectures for learning low-level sensory-motor behaviors. The purpose of designing a new architecture is to interface the low-level behavior learning and control which introduces an integration and behavior arbitration techniques by means of high-level control. Figure 1 depicts the proposed architecture. The major units are described below.

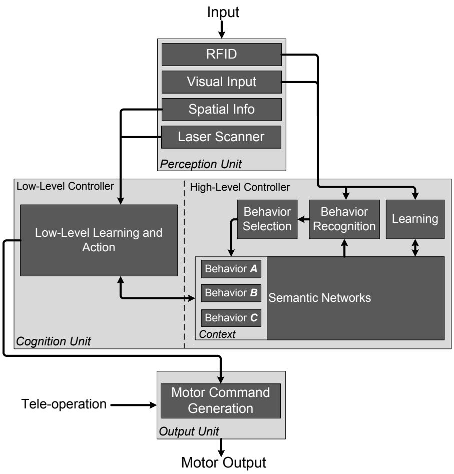

Figure 1: Proposed Architecture

2.1 Perception Unit

This unit collects and pre-processes sensor data from the environment. In our experimental set-up, the robot is equipped with laser scanner, ultrasonic transducers, infra red sensors and an RFID reader that acts as a high-level sensor which delivers identity (and to some extent position) of places, objects and people equipped with RFID tags. Every tag is associated with a number of properties defined in a database. The RFID technique is commonly used in robotics to get reliable performance and human-robot interaction (Nguyen et al., 2009), and should be considered as a complement to other high-level sensors like face recognizers, emotion detectors, gesture recognizers or any other visual inputs, and not a replacement.

2.2 Cognition Unit

The Cognition Unit is responsible for all robot decision making and action selection processes. It contains representation of the robot's cognitive state and has functions to modify its internal states.

Due to the structural differences between low and high-level information, the unit is organized in two modules running simultaneously.

2.2.1 High-Level Controller

One of the main tasks for this module is to learn contexts that are relevant for the execution of low-level behaviors. The other task is to arbitrate the low-level controller. The module relies on the abilities of a Semantic Network with predefined concepts and relations describing the environment. In learning mode, high-level inputs from the Perception unit update the SN such that contexts associated with the demonstrated behaviors are learned. In execution mode (performing phase), the module supports the low-level controller by activating the most relevant context(s) according to the current environmental conditions. These contexts act as bias in the activation of behaviors in the low-level controller. Basically, the cognitive state of the robot, represented in the Semantic Network, is updated through perception and a behavior recognition process, and acts as a cue for performing a behavior.

2.2.2 Low-Level Controller

The low-level controller learns and executes behaviors that are mappings of sensory-motor data to low-level actions (Billing et al., 2010a; Billing and Hellström, 2010b). In the presented work, the technique for learning is *Predictive Sequential Learning (PSL)* (Billing and Hellström, 2008). PSL treats control as a prediction problem and decides the next action based on the sequence on recent sensory-motor events. This technique allows learning of many types of complex behaviors, but does only work as long as the recent sensory-motor history contains all information necessary to select an appropriate action. One way to overcome this limitation is to define several contexts for PSL, where each context acts as a bias for action selection. In this way, actions that are less common within the present context are inhibited and the risk for selecting inappropriate actions is reduced.

2.3 Output Unit

This unit is designed to enable tele-operation of the robot. In addition, execution of action selection results coming from the Cognition unit into motor commands will be performed here.

3 SEMANTIC NETWORKS

Semantic Networks are often used to represent abstract knowledge in a human-like fashion. They are common within artificial intelligence as well as in philosophy, psychology and linguistics (Bagchi et al., 2000; Brown and Cocking, 2000; Rodriguez, 2008). In robotics, Semantic Networks is used for concept forming and situational awareness (Coradeschi and Saffiotti, 2003). The structured way of representing knowledge can in combination with visualization tools (Hartley and Barnden, 1997) help humans to understand the internal state of the robot and what is happening in the robot's cognitive system. This may for instance help a tutor to put the robot back on track when it is distracted during learning or performing phases.

In our usage of Semantic Networks, high-level concepts such as perceived object types and properties are represented as nodes while relations between concepts are represented as links. The initial SN is pre-defined and comprises nodes that are connected to the perception unit. These nodes are activated through perception.

3.1 Generalization Ability

A common reason for using a Semantic Network as a model of the environment is its ability to generalize (Mugnier, 1995), (Vashchenko, 1977). In our case, after a demonstration in LfD, the robot will be able to extend the learned context to other, related, contexts. Assume for instance that the robot learns how to clean the table if there are empty cups on it. By generalizing the cup concept to all the drinkwares, it will also perform the cleaning behavior when perceiving a mug on the table.

3.2 Interfacing to Low-Level Information

The success of robots designed to learn and work in daily environments with humans, relies on wrapping sensory-motor information with high-level concepts. This can improve human-robot interaction by utilizing Semantic Networks (Galindo et al., 2005). As mentioned earlier in section 2.2.2, *contexts* which are activated by the Semantic Network, give meaning to low-level information and act as a bias to choose suitable behaviors.

3.3 Spreading and Decaying Activation

In the proposed approach, each node has an activation level. *Spreading* is a mechanism by which activation spreads from one node to another in proportion to the strength of their connection. *Decaying* is a mechanism by which the activation levels of nodes gradually decrease over time. These processes have been implemented in a variety of ways to solve different problems in modeling, learning and robotics (Bagchi et al., 2000; Brown and Cocking, 2000). The spreading activation model used in this work, is based on mechanisms of human memory operations that originates from psychological studies (Rumelhart and Norman, 1983) and was first introduced in computing science in the area of artificial intelligence to provide a processing framework for Semantic Networks (Crestani, 1997). In order to make spreading activation work properly, we made following assumptions:

- Activation spreads in parallel, to all links leading out from a node
- Activation at a node is divided among the links that lead from it
- Activation decays rapidly without stimulation from other nodes or inputs
- Each node has an energy parameter that limits the number of link levels for spreading

The degree of generalization depends on the amount of energy available for propagation of activations. Higher amounts allow spreading along several links, which leads to higher connectivity of nodes and increase generalization.

The connections between nodes have weight values that limit the propagation of activation through the network. Learning is used to modify the connection weights and will be discussed in the next section.

4 LEARNING AND PERFORMING PHASES

One of the objectives of the research presented in this paper is to develop mechanisms to identify high-level contexts in a demonstration, and map each context to a low-level behavior. The low-level controller is assumed to contain learning capability based on sensory-motor data, and an ability to execute the behaviors on request. In section 4.1 we describe how high-level contexts are learned simultaneously with the low-level learning.

4.1 Learning Phase

Our learning approach is inspired by Novelty Detection techniques which are commonly used to detect new situations that did not occur during training (Markou and Singh, 2003).

We assume that we already have a predefined Semantic Network based on an ontology of the domain in which the robot should operate. This network is used to interface to the Perception unit and to identify or activate related nodes through spreading and decaying activation.

The learning process starts by generating a *History Network* describing the normal state of the world. The environment is observed by sampling the high-level sensors at a given frequency. As mentioned earlier, RFID tags are used for simplifying object detection and identification. Each readout gives object identities and properties that are perceived in the environment. Each read tag causes the corresponding nodes to be activated. For instance, if the RFID belonging to a red ball is detected, the nodes *Red* and *Ball* will be activated. Throughout the learning process, activation levels propagate to all connected nodes by spreading activation.

Sometimes a node is activated and deactivated due to noise and uncertainties in the RFID sensing. Therefore, a decaying delay parameter is defined to prevent instant deactivation of nodes after the disappearance of correspondent object from the environment.

Finally, the updated Semantic Network will be saved as the *History Network*.

Now learning of a high-level context may start. A *context node* with the name of the new behavior to be learned is added to the network. This version of the network is called *Learning Network*. The human teacher then demonstrates the wanted behavior by tele-operation. The RFID sensors perceive high-level concepts, at the same time as sensory-motor data is learned by the low-level controller. The context node will be connected to nodes activated by the RFID sensors. To finalize the learning process, two issues must be solved. First, the most relevant connections must be determined. Secondly, the weights between the remaining nodes and the context node must be computed. In order to identify relevant connections, the algorithm looks for significant differences between the *History* and *Learning Networks*. An unpaired T-Test is used to compare mean node activation for all nodes.

$$t_x = \frac{\mu_{AHx} - \mu_{ALx}}{\sqrt{\frac{Var_{Hx}}{n_H} + \frac{Var_{Lx}}{n_L}}} \tag{1}$$

where

μ_{AHx} is mean activation of History Node x
μ_{ALx} is mean activation of Learning Node x
n_H and n_L are number of samples for History and Learning respectively

t_x tells whether the samples for the two nodes are drawn from the same distribution or not. In other words: did the node change significantly between History and Learning phases. If it did not, the connection between the node and the context node should be removed. For instance, suppose *ambient light* was always *on*, during both *History* and *Learning* phases. In this case, the T-Test will consider *ambient light* as irrelevant because of the identical distribution in both phases.

After the elimination process of irrelevant connections, weights (w_x) for the remaining nodes are calculated. This is done by the following equation:

$$w_x = \frac{N_x \mu_{ALx}}{P} \tag{2}$$

where N_x is the number of samples for which node x has activation value above 0 during the learning phase, and P is the weighted sum for all nodes, calculated as follows:

$$P = \sum_{i=1}^{n} N_i \mu_{ALi} \tag{3}$$

Finally, the learned context node will be associated with the learned behavior representation in the low-level controller module.

4.2 Performing Phase

In the performing phase, RFID sensors update their corresponding network nodes. Whenever a node is activated, all other linked nodes are activated according to the spreading mechanism. In this way previously learned context nodes may get activated, thereby, guiding the low-level controller to execute the behaviors. If two or more contexts have high activation levels, several behaviors are possible, and the final decision will be made by the low-level controller. This can be viewed as high-level behavior recognition and is performed by *Behavior Recognition* module depicted in Figure 1. Due to the pre-defined semantic relations in the semantic network, the robot will be able to generalize the demonstrated context to similar contexts. As previously mentioned, the degree of generalization can be controlled by the amount of energy (Huang et al., 2006).

5 EXPERIMENTS

For better understanding of the whole approach, an example is shown. Assume we are going to teach the robot how to move a thing to the storage room. First, the robot will start moving around by tele-operation and collecting information regarding all the objects and places by RFID tags. Due to the characteristics of the described technique, the blue box should not be present at this stage. Figure 2 depicts the robot's perceptions that yield the *History Network*.

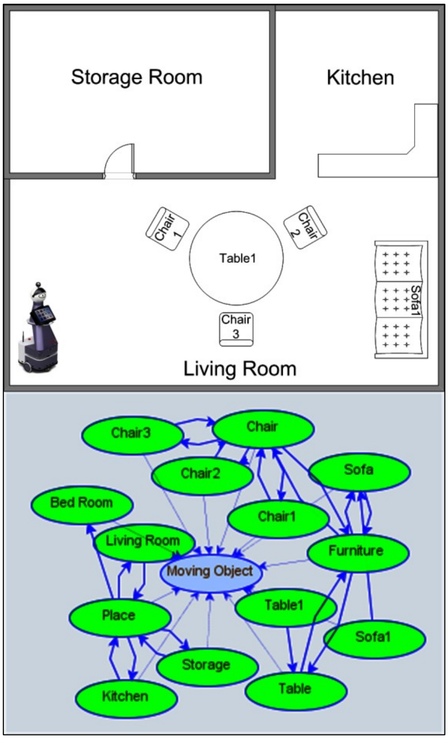

Figure 2: History Network

Learning will begin by placing the blue box somewhere in the middle of the room and tele-operating the robot towards the box. After grasping, the teacher commands the robot to push the box and guides it to the storage room that ends the learning phase. Figure 3 depicts the learned *Moving Object* behavior. Although we did not illustrate any low-level learning, this is done simultaneously by the low-level controller while tele-operating the robot.

The number of samplings for the history (N_H) and learning (N_L) is 40 and 60 respectively. In order to identify the nodes with the most significant changes, the t-test is run and results are shown in Table 1. Confidence Interval (CI) of the test is given by the t-distribution with α value set to 0.05. Degree of Freedom (DF) is calculated as follows:

$$DF = (N_H + N_L) - 2 \tag{4}$$

According to equation 1, t_x will be computed and nodes which fulfill condition 5 will be removed.

$$-CI \leq t_x \leq CI \tag{5}$$

Finally, according to equation 2, weights for the remaining nodes are calculated, shown in Table 1.
After finalizing the learning phase, the robot is able to perform *Moving Object* action whenever it perceives *blue* and *box1* in the environment.

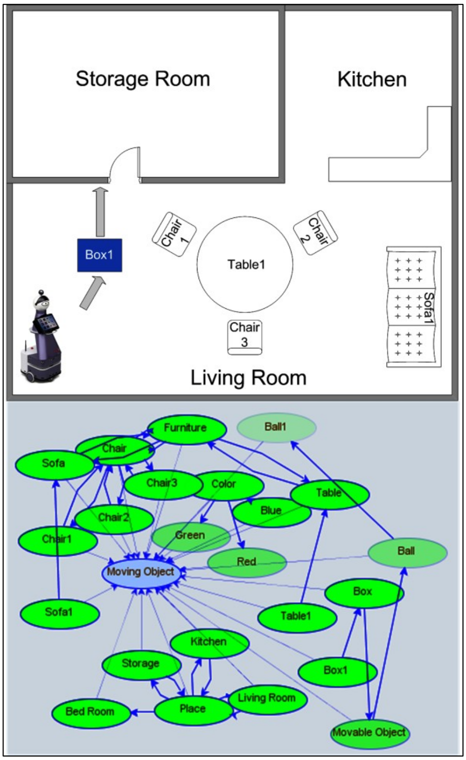

Figure 3: Learned Network

Table 1: History and Learning Values, T-Test results and Weight Values

Node	μ_{AHx}	σ_{Hx}	N_{Hx}	μ_{ALx}	σ_{Lx}	N_{Lx}	t_x	DF	CI	P_x	w_x
Living Room	1.51	0.56	36	1.65	0.37	58	-0.48	98	2.0	0%	--
Sofa	1.82	1.2	28	2.24	0.94	51	-1.0	98	2.0	0%	--
Table1	1.73	1.16	28	2.15	0.9	51	-1.03	98	2.0	0%	--
Furniture	2.64	1.62	31	3.21	1.24	54	-0.96	98	2.0	0%	--
Box	0.0	0.0	0	0.73	0.44	44	-6.52	58	2.02	19.74%	0.1974
Box1	0.0	0.0	0	0.73	0.44	44	-6.52	58	2.02	19.74%	0.1974
Bed Room	1.3	0.48	36	1.41	0.32	58	-0.48	98	2.0	0%	--
Chair2	2.81	1.71	31	3.41	1.31	54	-0.95	98	2.0	0%	--
Table	2.12	1.40	28	2.61	1.09	51	-1.0	98	2.0	0%	--
Blue	0.0	0.0	0	0.66	0.47	40	-6.21	58	2.02	16.32%	0.1632
Chair1	2.26	1.40	31	2.76	1.07	54	-0.97	98	2.0	0%	--
Kitchen	0.87	0.31	36	0.95	0.20	58	-0.46	98	2.0	0%	--
Movable Obj.	0.0	0.0	0	0.36	0.22	44	-6.52	58	2.02	9.87%	0.0987
Green	0.0	0.0	0	0.21	0.15	39	-6.14	58	2.02	5.17%	0.0517
Chair3	3.21	1.89	31	3.87	1.43	54	-0.93	98	2.0	0%	--
Sofa1	0.67	0.46	27	0.85	0.35	51	-1.09	98	2.0	0%	--
Ball1	0.0	0.0	0	0.09	0.05	44	-6.52	58	2.02	2.47%	0.0247
Red	0.0	0.0	0	0.22	0.15	40	-6.21	58	2.02	5.44%	0.0544
Storage Room	1.97	0.69	36	2.14	0.43	58	-0.46	98	2.0	0%	--
Ball	0.0	0.0	0	0.18	0.11	44	-6.52	58	2.02	4.94%	0.0494
Place	2.62	0.94	36	2.85	0.60	58	-0.46	98	2.0	0%	--
Chair	3.65	2.13	31	4.39	1.61	54	-0.92	98	2.0	0%	--
Color	0.0	0.0	0	0.66	0.47	40	-6.21	58	2.02	16.32%	0.1632

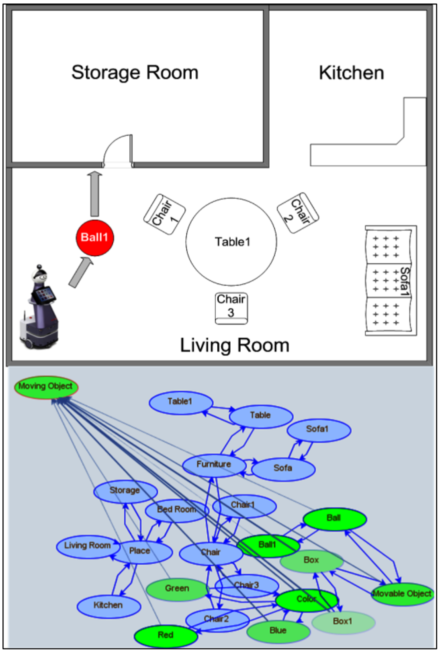

Figure 4: Performing Phase

The robot is not only capable of performing *Moving Object* behavior by observing the same objects during the learning phase, but can also generalize objects and concepts in the new situations. In order to test the system, the robot should recognize a *red ball* and push it to the storage room. This example clearly shows the generalization ability mentioned in section 3.1. As can be seen in Figure 4, perceiving *red* and *ball1* instead of *blue* and *box1*, to some degree, activates *Moving Object* context node through direct links and other connections to the *Color* and *Movable Object* nodes.

The activation level of the context node (A_c) is calculated by equation 6:

$$A_c = \sum_{i=1}^{n} A_i w_i \qquad (6)$$

n is the number of nodes which are currently activated and connected to the context node. A selection threshold should be defined for accepting the selected behavior as a result of generalization. In our example, we set the threshold to 0.6 meaning that the result of equation 6 should be at least 60% of the maximum value of the context node's activation (A_{cmax}). The maximum value is calculated during the learning phase by equation 6 and by replacing A_i with A_{imax} (maximum activation of node *i*). For this example, A_{cmax} equals 0.8214 and calculated A_c is 0.5246 which passed our threshold with 63%. Therefore, *red* and *ball1* are also able to trigger *Moving Object* context and cause the low-level controller to execute corresponding sensory-motor commands.

6 CONCLUSION AND FUTURE WORKS

In this paper we proposed an architecture to learn and act at a conceptual level by means of Semantic Networks. By introducing Semantic Networks and their usage in some research projects, a possible integration to LfD discussed. These aspects are valuable in concept forming and provide support for higher level cognitive activities such as behavior recognition. This integration is useful not only for LfD, but can be utilized in scaffolding, reinforcement learning or any other supervised learning algorithms. In this work, functionality of the system is tested with limited objects in the environment. In case of scaling up the number of entities in the working ontology, generalization will be more applicable.

Currently, our approach is incapable of handling quantities and negations. In our future work, we are going to define new link types in the Semantic Networks and design the high-level control in a way that can learn more complex scenarios.

ACKNOWLEDGEMENTS

This work has been financed by the EU funded Initial Training Network (ITN) in the Marie-Curie People Programme (FP7): INTRO (INTeractive RObotics research network), grant agreement no.: 238486.

REFERENCES

Bagchi, S., Biswas, G., and Kawamura, K., 2000. Task Planning under Uncertainty using a Spreading Activation Network, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 30, No. 6, pp. 639-650.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S., 2008. Robot programming by demonstration. In Siciliano, B. and Khatib, O., editors, *Handbook of Robotics*. Springer.

Billing, E. A., Hellström, T., and Janlert, L. E., 2010a. Behavior Recognition for Learning from Demonstration, *Proceedings of IEEE International Conference on Robotics and Automation*, Alaska.

Billing, E. A., and Hellström, T., 2008. Behavior Recognition for Segmentation of Demonstrated Tasks, *IEEE SMC International Conference on Distributed Human-Machine Systems*, pp. 228 – 234, Athens, Greece.

Billing, E. A. and Hellström, T., 2010b. A Formalism for Learning from Demonstration, *Journal of Behavioral Robotics*, Vol. 1, pp. 1–13.

Brown, A. L., and Cocking, R. R., 2000. *How people Learn: Brain, Mind, Experience, and School: Expanded Edition*, John D. Branford, National Academy Press.

Byrne R.W., Russon. A. E., 1998. Learning by imitation: a hierarchical approach, *The Journal of Behavioral and Brain Sciences*, Vol. 16.

Byrne, R. W., 1994. The evolution of intelligence. *In Behaviour and Evolution (ed. P. J. B. Slater and T. R. Halliday)*, pp.223-265. Cambridge University Press.

Coradeschi, S., Saffiotti, A., 2003. An Introduction to Anchoring Problem. *Journal of Robotics and Autonomous Systems*. Vol. 43, pp. 85-96.

Crestani, F., 1997. Application of Spreading Activation Techniques in Information Retrieval. *Journal of Artificial Intelligence Review*. Vol. 11, pp. 453-482.

Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., 2005. Multi-Hierarchical Semantic Maps for Mobile Robotics. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*. pp. 3492-3497.

Hartley, R. and Barnden, J., 1997. Trends in Cognitive Science, Vol. 1, No. 5, elsevier Science ltd.

Huang, Q. Y., Su, J. S., Zeng, Y. Z., Wang, Y. J., 2006. Spreading activation model for connectivity based clustering. *Advances in Information Systems*. Vol. 4243/2006, pp. 398-407.

Kasper, M., Fricke, G., Steuernagel, K., Puttkamer, E. V., 2001. A behavior-based mobile robot architecture for Learning from Demonstration. Journal of Robotics and Autonomous Systems. Vol. 34, pp. 153-164.

Kopp, S., Graeser, O,. 2006. Imitation Learning and Response Facilitation in Embodied Agents. *Proceedings of 6th International Conference on Intelligent Virtual Agents, Marina Del Ray, CA, USA*. pp. 28-41.

Markou, M., Singh, S., 2003. Novelty Detection: A Review – Part 1: Statistical Approaches. Journal of Signal Processing. Vol. 83, Issue 12, pp. 2481-2497.

Mugnier, M. L., 1995. On Generalization / Specialization for Conceptual Graphs. *Journal of Experimental & Theoretical Artificial Intelligence*. Vol. 7, Issue 3, pp. 325-344.

Nehaniv C. L., Dautenhahn K., 2000. Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. *Learning Robots: An Interdisciplinary*

Approach. Vol. 24, pp. 136–161, World Scientific Press.

Nguyen, H., Deyle, T., Reynolds M., and Kemp, C. C., 2009. PPS-tags: Physical, Perceptual and Semantic tags for autonomous mobile manipulation, *Proceedings of the IROS Workshop on Semantic Perception for Mobile Manipulation.*

Nicolescu, M. N., 2003. A framework for learning from demonstration, generalization and practice in human-robot domains. *Doctoral Dissertation*. University of Southern California, Los Angeles, CA, USA.

Rodriguez, M.A., 2008. Grammar-Based Random Walkers in Semantic Networks, *Knowledge-Based Systems*, Vol. 21, pp. 727-739.

Rumelhart, D., Norman, D., 1983. Representation in memory. *Technical Report*, Department of Psychology and Institute of Cognitive Science, UCSD La Jolla, USA.

Vashchenko, N. D., 1977. Concept Formation In a Semantic Network. *Cybernetics and System Analysis*. Vol. 19, No. 2, pp. 277-285.

Whiten, A., Ham, R., 1992. On the nature and evolution of imitation in the animal kingdom: Reappraisal of a century of research. *Advances in the Study of Behaviour,* Vol. 21, pp. 239-283.

II

Towards Goal Based Architecture Design for Learning High-Level Representation of Behaviors from Demonstration

Benjamin Fonooni, Thomas Hellström, and Lars-Erik Janlert

Abstract— **This paper gives a brief overview of challenges in designing cognitive architectures for Learning from Demonstration. By investigating features and functionality of some related architectures, we propose a modular architecture particularly suited for sequential learning high-level representations of behaviors. We head towards designing and implementing goal based imitation learning that not only allows the robot to learn necessary conditions for executing particular behaviors, but also to understand the intents of the tutor and reproduce the same behaviors accordingly.**

Index Terms— **Learning from Demonstration, Cognitive Architecture, Goal Based Imitation**

I. Introduction

LEARNING from Demonstration (LfD) is one of the most popular learning techniques to teach robots new skills by observing a human or robot tutor [2]. LfD involves several challenges, such as generalization of learned behaviors, representation of behaviors, sequence learning, and reproduction of complex behaviors [23], [24], [25]. Some researches proposed architectures based on biology and psychology of human or animal cognitive systems [3]. Examples of biologically inspired models are given by Billard et al. [7], Kopp et al. [8] and Demiris et al. [19]. The neural model approaches, fundamentally driven by mirror neuron systems [10], are also considered by many researchers [18], [19]. There are also some recent efforts in modeling goal-based imitation that infer intents of the tutor rather than repeating observed actions and following exact trajectories [4], [5], [16].

Most of the works referred above, focus on learning and reproduction of low-level representations (sensory-motor events) of behaviors. In this work we assume that these representations are already available as *behavior primitives* (below often referred to as *primitives*) such that no learning is required at the sensory-motor level. Primitives have been applied in robot control for several years, and there are proposed models describing challenges of connecting perception to primitives [13]. Primitives accomplish goal-directed behaviors and can be formalized as control policies [1]. Primitives may also represent complete temporal behaviors [20], [21].

The main goal of the work presented in this paper is to introduce a novel architecture for learning *contexts,* which are high-level representations of behaviors. Each context is associated with a predefined *action* and contains information on necessary perceptual conditions for this action to be executed. Actions are part of the architecture and act as interfaces between contexts and primitives in order to retrieve objects of attention from the contexts, convert them into low-level information and pass them as parameters to primitives.

We improve the previously developed architecture [6] by implementing cognitive mechanisms to learn intentions of the tutor and reproduce the behavior through activating learned contexts and recognizing the associated stimuli.

The remainder of the paper is structured as follows: In the next section, the proposed architecture and its components are elaborated. Section III introduces a novel algorithm for learning new contexts, and mechanisms for reproduction of learned behaviors. Section IV describes results from several experiments. Section V explains goal inference mechanisms which are key factors for behavior reproduction.

II. Architecture Overview

Fig. 1 depicts the developed architecture. In the following sections all units and modules are described.

A. Perception Unit

This unit is responsible for perceiving the environment by processing sensor data. Sensors can deliver either low-level data, like laser scanners, or high-level data, like gesture recognizers, emotion detectors and RFID tag readers. The difference between low and high-level data is the amount of processing required to connect the output of the sensors with concepts. For instance, reading the RFID tag of a cup and fetching its properties from a database, requires less processing than perceiving the cup and its properties solely by a laser scanner.

The Perception Unit delivers processed information to the various modules of the Cognition Unit.

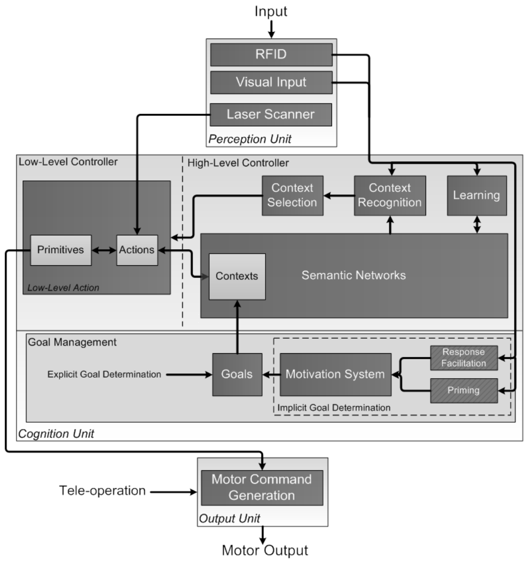

Fig. 1. Units and modules of the architecture

B. Cognition unit

The Cognition Unit consists of three main modules. The High and Low-Level controllers are responsible for learning, recognizing and executing contexts. The Goal Management module is designed for creating and inferring goals, explicitly and implicitly, as well as keeping track of current goal and intentions.

High-Level controller

This module is responsible for learning contexts. A main component of this module is the long-term memory represented by an, initially predefined, Semantic Network (SN). It contains nodes representing concepts and objects according to a pre-defined ontology. Each context is a node in the SN. By the learning process, it gets associated with a behavior primitive, and with concepts and objects in the SN. In our previous work we proposed a novel approach that creates connections (links) between the context and nodes connected to perception [6]. Perceptions are outputs of the perception unit, and activate the corresponding nodes in the SN. Each node has an activation level that defines how strongly it is activated.

After the learning phase, the robot should be able to recognize conditions for triggering a specific context, and thereby executing the associated behavior. This is denoted the reproduction phase. The perception mechanisms will change activation levels of sensed nodes, which in turn are connected to one or more contexts. Due to the spreading activation mechanisms in the SN, activation will be propagated to connected nodes [11] such that the robot will be able to generalize the learned contexts stored in the SN [12]. In this way, the Context Recognition module selects one or several contexts.

The most highly activated context will be selected by the Context Selection module and made available to the Low-Level Action Controller.

Low-Level controller

As mentioned earlier, the robot is equipped with a set of pre-defined behavior primitives. In the learning phase, the High-Level Controller associates a behavior primitive with the newly learned context via actions. This association is automatically recognized by the robot during tele-operation. In the reproduction phase, the Low-Level Controller is responsible for selecting motor commands in accordance with the pre-defined scheme in the selected primitive, and passing them to the Output Unit for execution.

a) Primitives

As described above, primitives are pre-defined low-level representations of behaviors. In the examples in this paper, we use “Grip” (gripping an object with the robot arm) and “Go to Location” (moving the robot to a particular location). The associations between actions and primitives are pre-defined.

b) Actions

Actions have an intermediary role in connecting contexts with primitives. They retrieve objects of attention from contexts and convert them to parameters required by the primitives. For instance, if the context “Get the Cup” is activated by perceiving the “Cup 1” as an object of attention, the associated action will pass “Cup 1” as parameter to the “Go to Location” primitive. The concept *Object of attention* refers to an object that the robot is going to work on. In the examples in this paper, we use three actions, each one mapped to a primitive. “Explore and Reach” is mapped to “Go to Location”, “Grab” is mapped to “Grip”, and “Move to Safe Location” is also mapped to “Go to Location”. The reason to keep actions and primitives separated is due to the possible association of several actions to one primitive. This allows actions to have different sets of conditions and way of providing parameters while primitives are only focusing on low-level aspects.

All actions and primitives are pre-defined in the system. The necessary pre-defined actions and primitives depend on the scenario and more importantly, the robot’s capabilities.

Goal Management

For most complex behaviors, several primitives have to be activated in sequence. A goal represents a sequence of contexts. Each goal has a set of conditions and objects of attention that defines what to look for and when to activate a specific context. Such a goal can be *“Help human rescue a victim”*, *“Moving victim to a safe place”* and etc.

One of the advancements of the current design in comparison with the previously developed architecture [6] is the Motivation system. It may be triggered by cognitive mechanisms such as response facilitation and priming, which motivate the robot to choose specific goals and eventually execute desired actions. This is denoted as implicit goal determination. Response facilitation is the phenomenon when observing a specific act, which is already in the repertoire of the robot, increases the probability of the robot later performing the same act [9]. One example is if the robot observes a human approaching a cup and grabbing it. This is identified as "Grab the Cup" behavior and increases the probability of the robot executing the same behavior.

Priming can be defined as an implicit memory effect that speeds up the response to stimuli because of exposure to a certain event or experience [22]. In our case, priming is the pre-activation of concepts stored in SN, in order to bias the learning process or affect the goal selection mechanism. For instance, if a "Red Ball" is shown to the robot, the nodes "Red" and "Ball" are primed and pre-activated. Therefore, the chances to select and satisfy goals that have connections to "Red" or "Ball" increase.

C. Output unit

This unit converts low-level commands from the Cognition unit to motor commands. In addition, it enables tele-operation of the robot.

III. Learning And Reproduction Phases

In the learning phase, a demonstration of a desired behavior is used to associate high-level contexts with perceived information. Each context is also mapped to a behavior primitive via an action such that the primitive will be executed when similar perception occurs during the reproduction phase.

A. Learning

The learning process is one of the main tasks of the architecture. In our previous work we have developed a learning algorithm based on novelty detection technique [6]. In this paper we will describe a new context-learning algorithm called *Multiple Demonstrations* (MD). We assume that we already have a number of predefined primitives and a predefined SN based on an ontology of the domain in which the robot should operate. The SN is interfaced to the Perception unit and activates related nodes through spreading and decaying activation mechanisms [11].

Context creation

The learning process starts by a tutor demonstrating the wanted behavior through tele-operation. A new context node is added to the SN. The robot observes the environment by sampling sensors at a given frequency. In the reported experiments, RFID tags are used for simplified object detection and identification. Each read-out gives identities and properties of objects perceived in the environment and causes the corresponding nodes to be activated. For instance, if the RFID belonging to a red ball is detected, the nodes "Red" and "Ball" will be activated. In this way, the RFID reader emulates sensors for object type and color. Throughout the learning process, activation levels propagate to all connected nodes by spreading activation. This mechanism allows the robot to generalize one concept to another. For controlling the degree of generalization, we define an energy level variable for each node. The energy level of a node determines how far activation level will spread from the initial node [15].

Sometimes nodes are deactivated during the demonstration due to noise and uncertainties in the RFID equipment. Therefore, a decaying delay parameter is defined to prevent immediate deactivation of a node when the corresponding object is not longer perceived in the environment.

The same behavior must be demonstrated to the robot at least twice. A new SN will be created each time, and the context node will be connected to nodes activated by the RFID read-outs. Due to noise and varying external conditions, these nodes may differ between demonstrations. To finalize the learning process, two issues must be solved: First, the most relevant connections must be determined. Second, suitable weights between the remaining nodes and the context node must be computed. In order to identify relevant connections, the MD algorithm looks for nodes with similar activation levels in all demonstrations. One-Way ANOVA [14] is used to compare mean node activation values of all nodes. The null hypothesis is that there is no significant difference between demonstrations for activation of a node. The following computations are performed for each one of the nodes connected to the new context node.

For each demonstration, sum of activations (S_{A_x}), activation mean value (μ_{A_x}), squares of deviations (d^2) and sum of squares of deviations (S_{d^2}) are calculated for each node:

$$d^2 = (A_x - \mu_{A_x})^2 \tag{1}$$

where A_x is the activation value of node x.

$$S_{d^2} = \sum A_x{}^2 - \frac{(\sum A_x)^2}{n} \tag{2}$$

where n is number of samples in the demonstration.

Grand Total (GT) is calculated as

$$GT = \sum S_{A_x}. \tag{3}$$

Then we calculate *total sum of squares (T)* as

$$T = GT - \frac{(GT)^2}{\sum_{i=1}^{r} n_i} \tag{4}$$

where r is the total number of demonstrations and n_i is the number of samples in demonstration i. *Between groups sum of squares (BG)* is calculated as

$$BG = \sum_{i=1}^{r} \frac{(S_{A_{x_i}})^2}{n_i} - \frac{(GT)^2}{\sum_{i=1}^{r} n_i}. \tag{5}$$

Within groups sum of squares (WG) is calculated as

$$WG = GT - \sum_{i=1}^{r} \frac{(S_{A_{x_i}})^2}{n_i}. \tag{6}$$

The number of degrees of freedom for *between groups sum of squares (BDF)* is calculated as

$$BDF = r - 1. \tag{7}$$

The number of degrees of freedom for *within groups sum of squares (WDF)* is calculated as follows:

$$WDF = r\left(\frac{\sum_{i=1}^{r} n_i}{r} - 1\right). \tag{8}$$

The *total degree of freedom (TDF)* is calculated as follows:

$$TDF = WDF + BDF. \tag{9}$$

Finally the F value is calculated as

$$F = \frac{\frac{BG}{BDF}}{\frac{WG}{WDF}}. \tag{10}$$

The F distribution (p=0.05) with the given BDF and WDF is then looked up. If the calculated F has higher value, we reject the null hypothesis and conclude that there is a significant difference between demonstrations for activation of the node. The node is then disconnected from the context node. The process is repeated for all nodes initially connected to the context node. Finally, weight values for the remaining nodes are calculated as

$$w_x = \frac{\sum_{i=1}^{r} N_{x_i} \mu_{A_{x_i}}}{P} \tag{11}$$

where N_{x_i} is the number of samples for which node x has activation value above 0 during the ith demonstration, and P is the weighted sum for all nodes, calculated as

$$P = \sum_{i=1}^{r} \sum_{j=1}^{n} N_{j_i} \mu_{A_{x_{j_i}}}. \tag{12}$$

After the process of context forming, the goals are created and related contexts are associated to each one of them.

Goal creation

The purpose of designing a goal based architecture is to help the robot identifying the intentions of the tutor. A goal is a sequence of contexts that represents a complex behavior. Fulfilling a goal means reproducing the sequence of corresponding primitives according to certain *conditions* set by the actions. Some of these conditions can be inferred from the predefined SN and are learned during the context learning, while the rest are hard-coded in the action associated with the learned context. As an example depicted in Fig. 3, "farness" of an object in "Explore and Reach" action cannot be inferred from the SN since its value changes by each sensor read-outs. Therefore, such a dynamic parameter cannot be represented as a node in the SN. Thus, part of the condition must be hard-coded to always check if the robot has sufficient distance to the object of attention in order to continue execution of the action. The relations between goals, contexts, actions and primitives are illustrated in Fig. 2 and elaborated in the next section.

In order to create a new goal, one has to break down a complex behavior into a set of contexts such that each one represents a behavior primitive. Due to the architectural design, each context maps to one action and each action maps to one behavior primitive. Therefore, complex behaviors are broken down into parts that can be executed by single predefined actions. Each such part is learned as a context. This is done by matching the tele-operation commands during demonstration with hard coded primitives and actions. After finishing the learning process of one context, the tutor starts demonstrating the next context. Environmental conditions help the robot to automatically learn the subsequent context as a sequence of the preceding one.

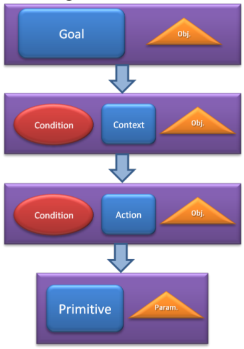

Fig. 2. Relations between Goal, Context, Action, and Primitive

One of the main assumptions is that actions and behavior primitives have a set of pre-defined parameters which applies to objects of attention. Both contexts and actions have objects of attention, which determine on which object to operate. All conditions defined in the actions are checked with the objects of attention.

As long as the conditions are still satisfied, the associated behavior primitive will be executed.

After completion of the high-level learning phase, all learned contexts and their corresponding actions are put together in a sequence, and a new goal object is created. New goals with associated contexts and corresponding actions are stored into a database for retrieval during the reproduction phase.

IV. Experiments

In this section we will present experimental results for learning and reproduction of new contexts. Consider, as an example, the behavior *"Take the Rubble from Human"* as part of an Urban Search and Rescue (USAR) application. The setting is a commercial/residential urban environment damaged by a severe earthquake. The goal for the robot is to assist a human agent cleaning a pile of rubbles covering a victim. The behavior starts with looking for a human agent, getting close to him/her, taking the rubble offered by the human, turning away and reaching the white sign (safe place). Fig. 3 depicts the *"Take the Rubble from Human"* goal, which shows the relations between contexts, their corresponding actions and objects of attention. The rest of the section explains how the robot can learn each context and reproduce the same behavior by perceiving similar environmental conditions.

The behavior is broken down into sub-behaviors such that each context can be associated to one of the pre-defined actions and primitives. It is the responsibility of the tutor to conduct the learning of each context in such a way that it can be associated with an action. The robot starts learning the first context "*Find Human"* as explained in Section III. This is illustrated in the left-most column in Figure 3.

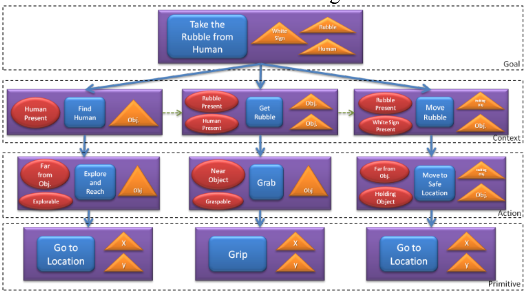

Fig. 3. Structure of *"Take rubble from human"* goal

The "Human Present" condition refers to a node connected to a high-level RFID sensor for detection of humans. Most of the time, an object is detected within a fraction of a second. Therefore, by executing *"Explore and Reach"* action frequently via tele-operation until detecting a human, the robot will establish connections between the *"Find Human"* context node and objects perceived by the RFID tag reader. The tutor must tele-operate the robot until the correct conditions for each context is learned. The pre-defined SN used for learning all contexts is shown in Fig. 4. Some nodes represent concepts and are denoted *category* nodes, the rest represent real objects in the world and are simply denoted *nodes*.

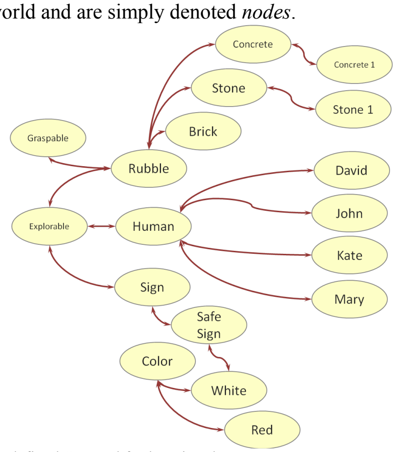

Fig. 4. Predefined SN used for learning the contexts

For learning the first context, *"Find Human"*, the robot recognizes "John" by the RFID tag on his bracelet during the exploration. As a result, the *"John"* node in the SN is activated and will spread the activation to the connecting nodes.

The tutor tele-operates the robot to get close enough to "John" and stops the robot. This signals that learning of the first context is completed. Fig. 5 shows the activation levels of all nodes during the learning of *"Find Human"* context. The behavior has been demonstrated four times with the same person and objects.

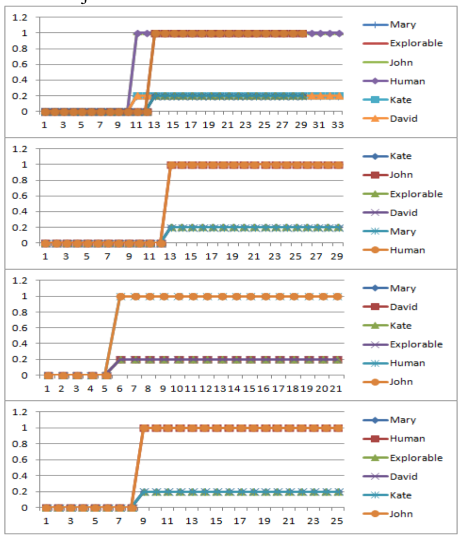

Fig. 5. Node activation levels for learning the *"Find Human"* context perceived in four demonstrations

Based on equations (1) to (10), results are calculated and shown in Table I.

TABLE I
"FIND HUMAN" CONTEXT LEARNING VALUES

Node	BG	WG	T	BDF	WDF	TDF	Cal. F
John	0.403	23.25	23.65	3	104	107	0.6017
Mary	0.016	0.93	0.94	3	104	107	0.6018
Kate	0.016	0.93	0.94	3	104	107	0.6018
David	0.016	0.93	0.94	3	104	107	0.6018
Human	0.403	23.25	23.65	3	104	107	0.6017
Explorable	0.016	0.93	0.94	3	104	107	0.6018

For all nodes, the calculated F-ratio is less than the tabulated value for the F-distribution at significance level $p=0.05$ (2.688), which means that node activations from different demonstrations are from the same distribution, and all nodes should remain connected to the context node.

The weight values for connecting nodes are calculated with equations (11) and (12) and are shown in Table II.

TABLE II
"FIND HUMAN" WEIGHT VALUES

Node	Weight
John	0.403
Mary	0.016
Kate	0.016
David	0.016
Human	0.403
Explorable	0.016

The final relations for the *"Find Human"* context are shown in Fig. 6. The solid links are semantic relations that come from the pre-defined SN and the dashed links are learned during the demonstration.

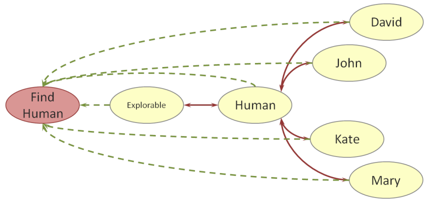

Fig. 6. "Find Human" context

Also features, like *"Explorable"* or *"Graspable"*, are represented as nodes in the SN. This facilitates reasoning and allows the system to check if an object of attention can satisfy conditions of the actions.

The learning process of the second context, *"Get Rubble"*, will start after robot reaches "John". "John" picks up a piece of stone (Stone1) and offers it to the robot, while the tutor grabs the piece with the robot arm through tele-operation. While learning this context, the *"Grab"* action command and its linked primitive are executed by the tutor so the robot can recognize the action and associate it with the context.

Through spreading activation, the concepts *"Stone"* and *"John"* will be generalized to *"Rubble"* and *"Human"* respectively. Therefore, the final SN does not only contain the objects perceived, but also similar objects.

For the *"Get Rubble"* context, the tutor demonstrated the behavior four times with the same objects and person. The learning values are listed in Table III.

TABLE III
"GET RUBBLE" CONTEXT LEARNING VALUES

Node	BG	WG	T	BDF	WDF	TDF	Cal. F
Concrete	0.004	0.264	0.268	3	82	85	0.429
Stone	0.323	20.60	20.93	3	82	85	0.429
Rubble	0.104	6.617	6.720	3	82	85	0.429
Brick	0.004	0.264	0.268	3	82	85	0.429
Stone1	0.323	20.60	20.93	3	82	85	0.429
Human	0.051	6.487	6.538	3	82	85	0.215
David	0.002	0.221	0.223	3	82	85	0.271
John	0.054	5.526	5.581	3	82	85	0.271
Mary	0.002	0.221	0.223	3	82	85	0.271
Kate	0.002	0.221	0.223	3	82	85	0.271
Graspable	0.004	0.264	0.268	3	82	85	0.429
Explorable	0.004	0.645	0.650	3	82	85	0.201

The tabulated value F (p=0.05) is 2.715, which means that all node activations are from the same distribution and as for the previous context, all nodes should remain connected. The weight values for remaining nodes are shown in Table IV.

TABLE IV
"GET RUBBLE" WEIGHT VALUES

Node	Weight
Concrete	0.0149
Stone	0.1315
Rubble	0.0745
Brick	0.0149
Stone1	0.1315
Human	0.2241
David	0.0421
John	0.2104
Mary	0.0421
Kate	0.0421
Graspable	0.0149
Explorable	0.057

As illustrated in Fig. 7, the *"Get Rubble"* context gets connected to two categories: *Human* and *Rubble*. In the action layer shown in Fig. 3, the *"Grab"* action requires an object that is close and graspable. Thus, the only category that meets this requirement is the *"Rubble"*. Therefore, all the conditions set in the *"Grab"* action are applied to objects in the *"Rubble"* category.

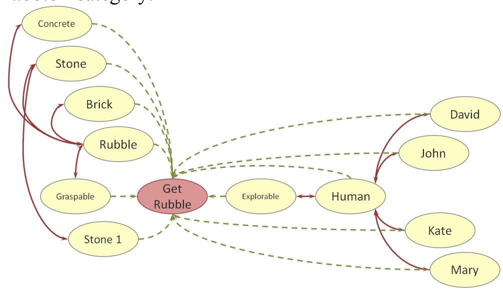

Fig. 7. "Get Rubble" context

The third and the last context to be learned is *"Move Rubble"* which starts when the robot holds the stone. The tutor tele-operates the robot to turn away from "John" and searches for the "Safe Sign". After reaching the designated location, demonstration of the whole behavior is completed.

The same computation is done for the *"Move Rubble"* context after four demonstrations. The computed values are listed in Table V.

TABLE V
"MOVE RUBBLE" CONTEXT LEARNING VALUES

Node	BG	WG	T	BDF	WDF	TDF	Cal. F
Concrete	0.003	0.878	0.882	3	88	91	0.105
Stone	0.042	17.75	17.79	3	88	91	0.07
Rubble	0.005	6.661	6.667	3	88	91	0.026
Brick	0.003	0.878	0.882	3	88	91	0.105
Stone1	0.069	14.4	14.47	3	88	91	0.141
Sign	0.305	23.83	24.13	3	88	91	0.376
Safe Sign	0.069	14.4	14.47	3	88	91	0.141
Color	0.069	14.4	14.47	3	88	91	0.141
White	0.069	14.4	14.47	3	88	91	0.141
Red	0.141	5.565	5.706	3	88	91	0.743
Graspable	0.003	0.878	0.882	3	88	91	0.105
Explorable	0.065	7.166	7.232	3	88	91	0.267

The tabulated value F (p=0.05) is 2.708, which is larger than all calculated F. Therefore, all nodes remain connected also for this context. The weight values for remaining nodes are shown in Table VI.

TABLE VI
"MOVE RUBBLE" WEIGHT VALUES

Node	Weight
Concrete	0.0299
Stone	0.1652
Rubble	0.0968
Brick	0.0299
Stone1	0.1503
Sign	0.1091
Safe Sign	0.1056
Color	0.0853
White	0.0853
Red	0.0426
Graspable	0.0299
Explorable	0.0696

What the robot has learned as *"Move Rubble"* is illustrated in Fig. 8.

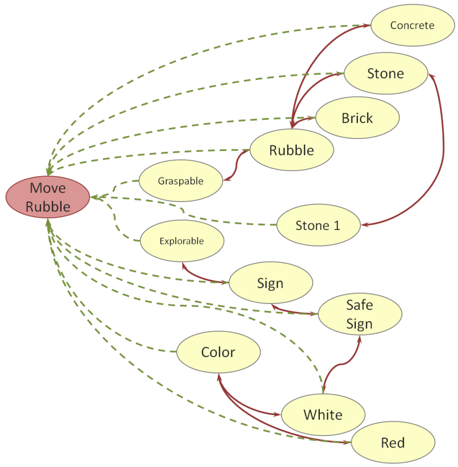

Fig. 8. "Move Rubble" context

Finally, all learned contexts are grouped together automatically to form a new goal that represents the demonstrated behavior (Fig. 3). The sequencing is done automatically by the learning routine that sets the start of the subsequent context to be the end of the preceding one.

V. Goal Inference

During the reproduction phase, the robot pursues a goal that is implicitly or explicitly determined by the user. In both cases, the robot attempts to infer what goals to pursue and activate related contexts in sequence in order to reproduce the learned behavior.

A. *Implicit goal determination*

Implicit goal determination is a bottom-up approach. Perceived objects or concepts activate contexts which in turn activate connected goals. The highest activated goal is selected. The motivation system plays a key role in implicit goal inference by putting the robot into different tracks by stimulating it with cognitive activities such as priming and response facilitation.

Continuing with the USAR scenario, suppose we prime the robot by showing a piece of concrete. As described earlier, concrete was not used directly in teaching but due to the generalization mechanism, contexts with connections to concrete or rubble in general will be activated. Goals connected to these contexts will be determined and listed. The goal connected to the highest activated context will be selected and the actions associated with the first context will be executed. Fig. 9 illustrates the goals and the effect of priming on selecting which goal to execute. In this example, the goal *"Take the Rubble from Human"* is selected.

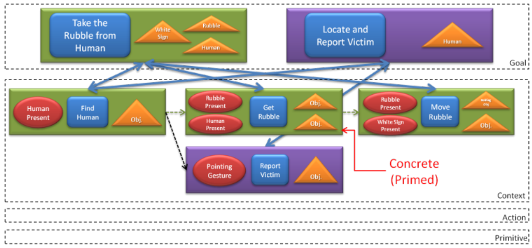

Fig. 9. Implicit goal determination; Priming causes the green goal and contexts to be selected for execution.

The reason is the priming effect which activates both *"Get Rubble"* and *"Move Rubble"* contexts. The activation levels of both contexts totally depend on environmental conditions and perception, but both belong to the *"Take the Rubble from Human"* goal. After determining the goal, the robot begins strolling around and perceiving the environment to fulfill the "Human Present" condition for the first context, *"Find Human"*. Then, it starts executing the action assigned to the context and its corresponding primitive until the action conditions ("Near Object" or "Explorable") are no longer satisfied. This means that the robot is able to find the human. Now, the second part of the sequence is selected and executed. The robot strolls around again until objects of attention required by the second context are perceived. According to Fig. 9, there are two possible choices as a second context: *"Get Rubble"* and *"Report Victim"*. The latter does not belong to the *"Take the Rubble from Human"* goal. As a result, *"Get Rubble"* and its associated action will be executed until the conditions ("Near Object" or "Graspable") are no longer satisfied. At this stage, the robot will start the last part of the sequence and finally stop when it reaches the "white safe sign". In Fig. 9, the robot's choices are illustrated by green boxes and transitions between the contexts are shown with dashed green arrows. The actions and primitives are shown in Fig. 3.

B. *Explicit goal determination*

Beside implicit goal determination, the user may explicitly specify a goal for the robot. The robot will then select only the contexts that fulfill the specified goal. For instance, if a user specifies *"Take the Rubble from Human"* as goal, the robot will only check for relevant objects and select contexts that are part of the specified goal. Thus, the robot will work top-down to identify the first context of the goal and as a result look for objects of attention defined by the *"Find Human"* context. Depending on the current state of the robot and environment, it may skip executing the first context if it has already reached a human. Thus, it checks for the conditions defined by the *"Get Rubble"* context. The process of context activation and action reproduction continues until the whole sequence is completed. Fig. 10 illustrates the mechanism for explicit goal determination. The green boxes and numbers show how the robot manages the sequential execution of contexts to achieve the selected goal.

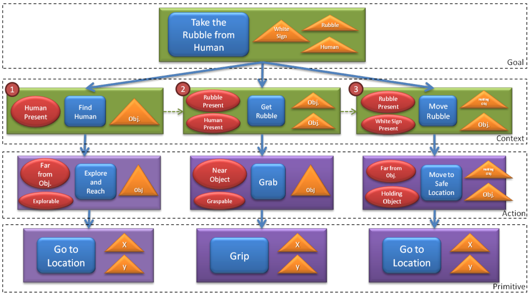

Fig. 10. Explicit goal determination

One of the main strengths of the presented design is the automatic execution of contexts in the right order. This is made possible by the learned conditions that guide the robot to do the right thing at the right time.

VI. Conclusion

In this paper we outlined an architecture for Learning from Demonstration. Considering strengths and weaknesses of other architectures, we proposed a new design for learning high-level representation of the behaviors and associating them with behavior primitives. The modules of the architecture were elaborated and mechanisms for information flow discussed. The Multiple Demonstrations context learning technique was introduced and a mechanism to detect irrelevant nodes was elaborated.

In this research we headed for goal based imitation learning, and by introducing goal creation and inference mechanisms, the robot was able to recognize the tutor's intentions. With the help of the motivation system, the robot can reproduce learned behaviors and pursue specified goals. Finally, the procedure for setting explicit or implicit goals for the robot under the USAR application scenario was discussed.

Acknowledgment

This work was financed by the EU funded Initial Training Network (ITN) in the Marie-Curie People Programme (FP7): INTRO (INTeractive RObotics research network), grant agreement no.: 238486.

References

[1] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in Cognitive Sciences*, 3(6): pp. 233-242, 1999.

[2] A. Billard, S. Calinon, R. Dillmann and S. Schaal, "Robot programming by demonstration," *Handbook of Robotics*, B. Siciliano and O. Khatib ed., Springer, 2008.

[3] H. Tan, "Implementation of a Framework for Imitation Learning on a Humanoid Robot Using a Cognitive Architecture," *The Future of Humanoid Robots - Research and Applications*, R. Zaier ed., 2012.

[4] W. Erlhagen, A. Mukovskiy, E. Bicho, G. Panin, C. Kiss, A. Knoll, H. van Schie, and H. Bekkering. "Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning," *Robotics and Autonomous Systems*, 54(5): pp. 353-360, 2006.

[5] C. Chao, M. Cakmak and A. L. Thomaz, "Towards grounding concepts for transfer in goal learning from demonstration," *In proceedings of IEEE International Conference on Development and Learning (ICDL)*, vol. 2, pp. 1-6, 2011.

[6] B. Fonooni, T. Hellström and L. E. Janlert, "Learning high-level behaviors from demonstration through Semantic Networks," *In proceedings of 4th International Conference on Agents and Artificial Intelligence (ICAART)*, pp. 419-426, 2012.

[7] A. Billard and M. J. Mataric, "Learning human arm movements by imitation: evaluation of a biologically inspired connectionist architecture," *Robotics & Autonomous systems*, 37(2-3): pp. 145-160, 2001.

[8] S. Kopp and O. Graeser, "Imitation Learning and Response Facilitation in Embodied Agents," *In Proceedings of the 6th international conference on Intelligent Virtual Agents*, pp. 28-41, 2006.

[9] R. W. Byrne and A. E. Russon, "Learning by imitation: A hierarchical approach," *The Behavioural and Brain Sciences*, Vol. 21, pp. 667-721, 1998.

[10] G. Rizzolatti and L. Craighero, "The Mirror-Neuron System," *Annual reviews of neuroscience*, Vol. 27, pp. 169-192, 2004.

[11] F. Crestani, "Application of Spreading Activation Techniques in Information Retrieval," *Journal of Artificial Intelligence Review*, Vol. 11, pp. 453-482, 1997.

[12] M. L. Mugnier, "On Generalization / Specialization for Conceptual Graphs," *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 7, Issue 3, pp. 325-344, 1995.

[13] M. J. Mataric, "Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics," *Imitation in Animals and Artifacts*, C. Nehaniv and K. Dautenhahn ed., The MIT Press, 2001.

[14] D. C. Howell, "Statistical methods for psychology," *Pacific Grove*, 7th ed., CA, Duxbury, 2010.

[15] Q. Y. Huang, J. S. Su, Y. Z. Zeng, Y. J. Wang, "Spreading activation model for connectivity based clustering," *Advances in Information Systems*, Vol. 4243/2006, pp. 398-407, 2006.

[16] C. Breazeal, M. Berlin, A. Brooks, J. Gray and A. L. Thomaz., "Using perspective taking to learn from ambiguous demonstrations," *Robotics and Autonomous Systems*, 54(5): pp. 385-393, 2006.

[17] E. Oztop, M. Kawato and M. A. Arbib, "Mirror neurons and imitation: A computationally guided review," *Neural Networks*, 19(3): pp. 254-21, 2006.

[18] E. Sauser and A. Billard, "Parallel and distributed neural models of the ideomotor principle: An investigation of imitative cortical pathways," *Neural Networks*, 19(3):285-298, 2006.

[19] Y. Demiris and G. Hayes, "Imitation as a dual route process featuring predictive and learning components: a biologically plausible computational model," *Imitation in Animals and Artifacts*, C. Nehaniv and K. Dautenhahn ed., pp. 327-361, MIT Press, 2002.

[20] C. L. Nehaniv, "Nine billion correspondence problems and some methods for solving them," *In Proceedings of the International Symposium on Imitation in Animals and Artifacts (AISB)*, pp. 93-95, 2003.

[21] D. Sternad and S. Schaal, "Segmentation of end-point trajectories does not imply segmented control," *Experimental Brain Research*, 124/1: pp. 118-136, 1999.

[22] J. Neely, "Semantic priming effects in visual word recognition: A selective review of current findings and theories," *Basic processing in reading: visual word recognition. Hillsdale*, D. Besner and G. Humphreys ed., NJ: Lawrence Erlbaum Association, Inc., pp. 264-336, 1991.

[23] E. A. Billing and T. Hellström, "A formalism for learning from demonstration," *Paladyn Journal of Behavioral Robotics*, 1 (1), pp. 1-13, 2010.

[24] E. A. Billing and T. Hellström, "Behavior recognition for segmentation of demonstrated tasks," *IEEE SMC International Conference on Distributed Human-Machine Systems*, pp. 228-234, 2008.

[25] E. A. Billing, T. Hellström and L. E. Janlert, "Behavior recognition for learning from demonstration," *International Conference on Robotics and Automation (ICRA)*, pp. 866-872, 2010.

III

Applying Ant Colony Optimization algorithms for High-Level Behavior Learning and Reproduction from Demonstrations

Benjamin Fonooni[1], Aleksandar Jevtić[2], Thomas Hellström[1], and Lars-Erik Janlert[1]
[1]Department of Computing Science, Umeå University, Sweden
{fonooni, thomash, lej}@cs.umu.se

[2]Robosoft S.A, France
aleksandar.jevtic@robosoft.com

Abstract

In domains where robots carry out human's tasks, the ability of learning new behaviors easily and quickly plays an important role. The idea behind Learning from Demonstration (LfD) is to identify what is the important information in the demonstrated behavior that requires remarkable attention from the robot. Furthermore, generalizing the learned behavior such that the robot is able to exhibit the same behavior under novel and unseen situations is another big challenge.

The main goal of this paper is to incorporate Ant Colony Optimization algorithms into LfD in an approach which focuses on understanding tutor's intentions, purpose of demonstrations and learning conditions to exhibit a behavior. The proposed method combines Ant Colony Optimization algorithms with semantic networks and spreading activation mechanism to reason and generalize the knowledge obtained through demonstrations. The approach also provides structures for behavior reproduction under new circumstances. Finally, applicability of the system in an object shape classification scenario is evaluated.

1. Introduction

During the past years robot task learning has received remarkable attention and motivated the robotics community to take a deeper interest in a technique based on human skill learning from observation (Billard et al., 2008). In robotics, such an approach fits in the framework of Learning from Demonstration (LfD). LfD is a promising way to naturally and intuitively teach robots new behaviors (skills) by demonstrating how to achieve the behavior (Argall et al., 2009). Applying LfD does not require any expert knowledge of robotics or domain dynamics, so it can easily be applied by non-roboticist users for straightforward or even non-trivial behaviors. While there are well known advantages to LfD, number of questions have to be answered in order to have true imitation which brought up to attention by Schaal

(Schaal, 1999) and Demiris et al. (Demiris and Hayes, 2002). These central questions are known as “Big Five” and include *“Who to imitate?”, “When to imitate?”, “How to imitate?”, “What to imitate?” and “How to evaluate successful imitation?”* (Dautenhahn and Nehaniv, 2002). Exhaustive overview of LfD and “Big Five” can be found here (Billard et al., 2008; Argall et al., 2009; Breazeal and Scassellati, 2002). Among those, *How* and *What* are mostly studied in the literature and approaches to face these challenges are from both high and low-level. The low-level is referred to mapping of sensory-motor information which produces actions that are performed by robot's actuators. Works by Mataric (Mataric, 2002), Dillmann (Dillmann, 2004), Ekvall et al. (Ekvall and Kragic, 2005), Pastor et al. (Pastor et al., 2009), Billing et al. (Billing and Hellström, 2010) and many others addressed the low-level perspective of LfD. The other aspect is referred to high-level and focuses on tutor's intention, goal of demonstration and to what objects, concepts or environmental states robot must direct its attention. In order to fully reproduce observed behaviors, understanding goals and results of actions are necessary. Otherwise, robot may produce the same motor actions, but they might have different effects to the world. Therefore, developing a sophisticated attention mechanism to identify the most important elements of demonstrations is essential. Works by Mahmoodian et al. (Mahmoodian et al., 2013), Hajimirsadeghi et al. (Hajimirsadeghi et al., 2012), Cakmak et al. (Cakmak et al., 2009), Erlhagen et al. (Erlhagen et al., 2006) and Chao et al. (Chao et al., 2011) addressed challenges of conceptualization and goal identification from demonstrations.

Under the LfD learning mechanism, first, tutor demonstrates a desired behavior to the robot so it produces mappings of sensory-motor states and generalizes the mappings to new examples. There are mainly two ways of conducting a demonstration for the robot (Ekvall, 2007):

1. *Direct-learning:* In this approach the tutor demonstrates a behavior directly by manually steering the robot using some devices such as a joystick.

2. *Indirect-learning:* In this approach the tutor enacts a behavior and the robot learns it by observation. Usually, vision and some other remote sensing methods are applied to record the demonstration. In this work indirect-learning has been applied and RFID sensing is used to record environmental states containing objects that appeared during demonstrations. This method is explained in detail in section 4.

Generalization is not only applied to low-level skills, but also in learning concepts and high-level representation of behaviors has a significant role. According to Mitchell et al. (Mitchell et al., 1986), generalization defined as a process of identifying common features from observing a set of training examples and forming a concept definition based on these features. We are interested in designing learning methods that are able to extend the knowledge from learned behaviors under known circumstances, to novel and unseen situations.

In the current paper, we address the *What* and *When* questions from high-level perspective while employing methods to learn and reproduce motor actions from demonstrations. Therefore, our proposed methods are only able to learn and reproduce high-level aspects of demonstrated behaviors. For this purpose, we utilized Semantic Network (SN) as a model to represent behaviors by nodes and linked them to a set of other nodes which are basically corresponding to concepts and objects in the real world. The network contains list of concepts, objects and their properties required for learning and reproduction of behaviors and must be provided prior to learning/reproduction process. The learned behaviors are then used as object affordances as well as preparing the ground for behavior arbitration. Our learning methods also provide techniques to learn conditions that result in the behavior and thus answer the question of *When* to imitate. These conditions can be environmental, objects to use, and concepts related to demonstrated behavior. Therefore, depending on the amount of knowledge available in the SN, robot may perceive enormous amounts of information during the learning. In so many cases when the desired behavior has a high degree of complexity or is demonstrated in an ambiguous manner, the robot requires a bias in order to focus on the right aspects of demonstration (Bensch and Hellström, 2010). By having a controller from a higher abstraction level to guide the robot especially during the learning phase, the robot's attention can be directed to aspects of demonstration that are significant for the behavior it is learning. This controller is a part of an architecture that has been proposed and developed in our previous works (Fonooni et al., 2012; Fonooni et al., 2013) and is employed in the current paper. However, we will not go through details of the used architecture.

Our learning method from previous paper (Fonooni et al., 2013) is based on one-way ANOVA test and has limitations due to the imposed statistical constraints. This prevents it from being successfully applicable in learning behaviors that are performed in noisy environments. Hence, to overcome this issue, our new method views the problem of behavior learning as an optimization problem and attempts to apply ACO algorithms to determine the most related elements of demonstrations.

Ant Colony Optimization (ACO) is a metaheuristic that has been used to solve numerous complex engineering problems that can be represented as discrete optimization problems (Dorigo and Stützle, 2002). ACO implements the pheromone-laying behavior that natural ant colonies use to store the information about the environment, which can then be locally accessed by any member of the colony. In most of the cases, the goal of applying the ACO is to find the shortest path between the points in the solution space or to extract the accumulated pheromone pattern (Jevtić, 2011).

This work proposes, as an original contribution, the use of semantic networks for biasing the robot and use of ACO algorithms to learn new behaviors and define degree of generalization in order to exhibit the learned behaviors in new and unseen situations.

The general design methodology for Swarm Intelligence tools proposed in (Jevtić, 2011) is used to develop the ACO-based learning algorithm. The proposed methodology consists of the following steps:

- Define the nodes that constitute the discrete data space in which the ants move.
- Define the set of application-specific variables and calculate probabilities of the displacement to the neighboring nodes.
- Apply the roulette rule as the underlying decision-making mechanism.
- Define the objective function for solutions evaluation.

The rest of the article is organized in the following manner: section 2 presents principal elements of Semantic Networks, section 3 presents the theory of ACO algorithms, section 4 presents the descriptions of learning and reproduction of high-level representation of behaviors, section 5 presents some of the tests and results, and finally section 6 presents conclusions.

2. Semantic Networks

The main concern in modeling the world is how to structure and generalize information. Semantic Networks (SN) are common techniques to represent abstract knowledge in systems based on artificial intelligence. In robotics, SN is used more often for concept forming, situational awareness (Coradeschi and Saffiotti, 2003) and task planning (Galindo et al., 2007). In concept forming, monitoring the robot's attentional state to understand the amount of resources being engaged for each element of the demonstration is essential for a tutor during the learning period. Therefore, SN has been used to help the tutor check whether the robot is focusing on the right elements or if it needs a bias to get back on the right track.

The initial semantic network is predefined and contains all necessary concepts and objects that the robot is able to work with. Fig. 1 is an example of such a network used in our experiments. In the current research, high-level concepts such as **object categories** ("Basket", "Cylindrical", ...) and **objects** ("C1", "B1", …) are represented as nodes while their relations are represented as links in the SN. Nodes are connected to their child nodes in both directions, and all bidirectional links are representing associations.

A learned behavior is also represented by node in the SN and in the scope of this paper named as *context*. The context is a node which is formed by the learning method due to robot's perceptions and contains associations to all objects, concepts and environmental states related to the demonstrated behavior.

A common reason for using SN as a model of the world is its ability to generalize information (Rogers, 2008). After each demonstration, the robot is able to extend the learned behavior under known circumstances to new, related situations. Assume that the robot learns how to clean the environment from a pile of bricks. By generalizing the brick concept to all kinds of rubble, it will also engage the cleaning behavior when observing a pile of concrete shards. The generalization is done by spreading activation which is a fundamental function in the SN (Crestani, 1997).

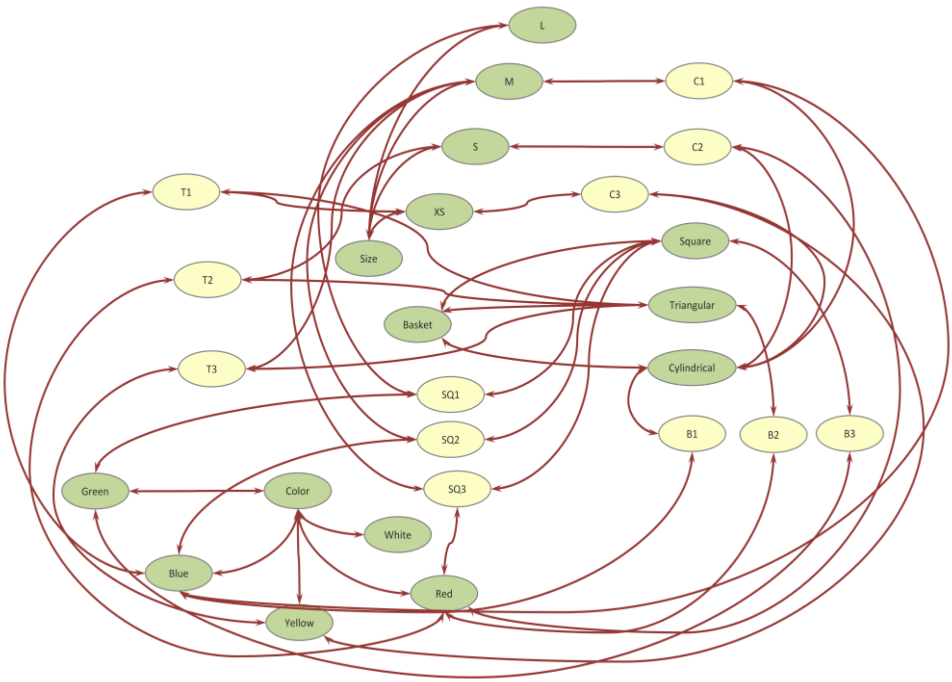

Figure 1. Example of predefined SN used in our experiments. Green nodes are representing concepts while yellow ones are representing objects.

2.1 Spreading Activation

The predefined SN is not informative per se, it requires a method to query the network and retrieve the information; this method is called *Spreading Activation*. The hierarchical network model is the base for long-term memory which contains interconnected nodes of information. The connections implement associations between the nodes and can control how to retrieve information.

When a node is activated by perceptual input, its activation value is set to 1.0, which is then propagated to its connections depending on **weight values, energy levels** and **decay factor**. We suppose all the links have initial weight values depending on the number of connections their parents have. For instance, according to Fig. 1, the *"Color"* node has five connections (*"Green", "Blue", "Yellow", "Red"* and *"White"*), thus connections from *"Color"* node to each of them have initial weight values equal to 0.2. Moreover, *"White"* has only one connection and that sets the initial weight value of the connection from *"White"* to *"Color"* to 1.0. This example also shows that due to having bidirectional links between nodes, weight values of two connected nodes are not necessarily identical.

The second parameter is the energy level that in combination with the decay factor controls how far activation propagates (Huang et al., 2006). These parameters control the degree of generalization, that is, how far the concept is extended to be less specific. As an example, “B1” is “Cylindrical” and “Cylindrical” is a “Basket”, so we can deduce that “B1” is a “Basket”. In other words, we generalize from “B1” which is highly specific to “Basket” which is less specific. The formulation for spreading activation used in this article is as follows:

$$a_j(t+\Delta t) = \begin{cases} a_j(t) + d\sum_i w_i a_i(t) & e_i > e_0 \\ a_j(t) & otherwise \end{cases} \quad (1)$$

where $a_j(t)$ is activation value of node *j* at time *t*,
$a_i(t)$ is activation value of node *i*, parent of node *j*, at time *t*,
Δt is duration of a time step,
$d \in [0,1]$ is decay factor,
and w_i is the weight value of the connection from node *i* to *j* and $w_i \in [0,1]$.
e_i is energy level of node *i* and calculated as follows:

$$e_i(t+\Delta t) = \begin{cases} e_i(t) + d\sum_n e_n(t) w_n & i \in C_n \\ 0 & otherwise \end{cases} \quad (2)$$

where $e_n(t)$ is energy level of parent of node *i* and $e_i(t) \in [0,1]$, C_n is a set consist of child nodes of node *n*.

e_0 is energy threshold, which is used to avoid firing activation in a loop. Since nodes can be connected in loops, firing activation from a node can run forever unless its energy is limited.

The decay factor (*d*) is a distance constraint that is used to control the degree of generalization. The rationale behind decay factor is that the strength of relation between two nodes decreases with their semantic distance (Crestani, 1997). Consequently, setting *d* close to 1.0 allows the system to generalize the context to the whole network while setting it to 0.0 results in more specific context.

2.2 RFID Sensing and Spreading Activation

Nodes are activated through perception of their corresponding object, person or location in the environment. The set of concepts, objects and their features are available as an ontology in the form of a predefined SN. Each object node is mapped to a RFID tag number and is detected by a RFID reader. By reading a tag number with the RFID reader, the associated object node and all its connected nodes will receive activation, meaning that the robot has perceived the object and recalled its features. As mentioned earlier, the number of nodes that the activation spreads to is proportionate to energy level of the parent node and the energy threshold.

There are also other techniques available to apply as a substitute to the RFID technique but require much more complicated technology, like image processing and vision techniques which are not concerns of the present research.

3. Ant Algorithms

The first successful Ant Colony Optimization (ACO) algorithm was introduced by Marco Dorigo who was inspired by biological works of Deneubourg and colleagues (Dorigo and Stützle, 2004). They proposed stochastic model of ant's behavior by observing ant colonies and how they are searching for the shortest path between food sources and their nest (Deneubourg et al., 1990). The algorithm simulates foraging behavior of Argentine ants, which is to explore the surrounding area randomly and leave a pheromone trail on the ground while moving. In case of finding food, on their way back to the nest, they will leave a trail of pheromone whose quantity depends on the quality of the found food. This will guide other ants to choose the path that leads to high quality food by tracing the paths with strong pheromone concentrations (Blum, 2005). This way of indirect communication between the ants is named *stigmergy* (Marsh and Onof, 2008). In order to allow the ants to always search for better solutions, negative feedback through pheromone evaporation is applied to restrain the ants from taking the same path.

In the following sections we are going to give an overview of Ant System (AS) and Ant Colony System (ACS) meta- heuristic algorithms based on our application of LfD using predefined SN.

3.1 Ant System (AS) Algorithm Description

ACO is a collection of meta-heuristic techniques in which the Ant System (AS) is the first algorithm proposed in the literature (Dorigo et al., 1996). An important question about AS technique is how it updates the pheromone values of the paths that are explored by all *m* ants. The pheromone value $\tau(r,s)$ associated with the edge between node *r* and *s* is updated as follows:

$$\tau(r,s) = (1-\rho).\tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s) \qquad (3)$$

where ρ is the pheromone evaporation rate,

m is total number of ants,

and $\Delta\tau_k(r,s)$ is the quantity of pheromone laid on edge *(r,s)* by ant *k* formulated as follows:

$$\Delta\tau_k(r,s) = \begin{cases} B_k & if\ (r,s) \in tour\ done\ by\ ant\ k \\ 0 & Otherwise \end{cases} \qquad (4)$$

$B_k = \frac{a}{L_k}$, L_k is number of levels traversed by ant k and a is activation value of starting node.

While building a solution, ants choose the next node according to a probability distribution:

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s)].[\eta(r,s)]^{\beta}}{\sum_{u \in J_k(r)} [\tau(r,s)].[\eta(r,s)]^{\beta}}, & if\ s \in J_k(r) \\ 0 & Otherwise \end{cases} \qquad (5)$$

where $J_k(r)$ is the set of feasible nodes containing edges *(r, l)* where *l* is a node not yet visited by ant *k*. Parameter β is used to control the relative importance of the pheromone versus the heuristic information $\eta(r,s)$, defined as:

$$\eta = \frac{1}{L} \qquad (6)$$

where L is number of intermediary connections from the starting node to the context node. The context node in SN represents the demonstrated behavior or sub-behavior that the robot is about to learn. It is set to be a goal for ants, so they leave more pheromone trails on a path that leads to the context node. The full explanation of AS usage in learning contexts is given in section 4.

3.2 Ant Colony System (ACS) Algorithm Description

The ACS is an extension to the Ant System algorithm. The principal differences between ACS and AS are changes in the node transition rule and the global pheromone updating rule (Dorigo and Gambardella, 1997). In ACS each ant builds a feasible solution to the goal node by applying the node transition rule (7) repeatedly. By passing each edge, an ant updates the amount of pheromone on the visited edges using the local pheromone update rule (8). After all ants have finished their tour, the amount of pheromone will be updated once more using the global pheromone update rule (9). This only gives the best ants, which were able to construct the best solution from the beginning of the trial, the chance to update the pheromone values. The node transition rule in ACS is formulated as follows:

$$S = \begin{cases} argmax_{u \in J_k(r)} \{[\tau(r,u)].[\eta(r,u)]^{\beta}\} & if\ q \leq q_0 \\ S & Otherwise \end{cases} \qquad (7)$$

where q is a random number uniformly distributed in [0,1], q_0 is a parameter ranging within $0 \leq q_0 \leq 1$, and S is a random variable selected according to probability distribution given in (5). The parameter q_0 determines the relative importance of exploitation versus exploration. A random number q is generated and according to (7), the best edge is selected (exploitation) or any edge will be selected according to (5). All other parameters are as same as the ones explained in AS algorithm. The local pheromone updating rule is calculated as:

$$\tau(r,s) = (1-\rho).\tau(r,s) + \rho.\tau_0 \qquad (8)$$

where τ_0 is the initial pheromone level and all parameters are as same as (3). The global pheromone updating rule is formulated as:

$$\tau(r,s) = (1-\rho).\tau(r,s) + \rho.\Delta\tau(r,s) \qquad (9)$$

where

$$\Delta\tau(r,s) = \begin{cases} \frac{a}{L_{gb}} & if\ (r,s) \in global\ best\ tour \\ 0 & Otherwise \end{cases} \qquad (10)$$

and L_{gb} is the minimum level of the globally best tour from the beginning of the trial.

4. Learning and Reproducing Behaviors

4.1 Learning

The proposed approach is aimed for teaching the robot the necessary **conditions** used for reproducing behaviors, and control the way it generalizes the learned conditions. The conditions are environmental states, presence and properties of perceived objects, and associations to other nodes in the semantic network. During the learning phase, the nodes representing the conditions are linked to a new node, denoted **context node**. What the robot perceives is linked to the context node directly and thus activation value of the perceived node spreads simultaneously to all its connections according to equations (1) and (2). All other activated nodes are conjointly linked to the context node as a result.

The learning phase starts by demonstrating the desired behavior several times with teleoperation. During each demonstration, the robot perceives objects in the environment with an RFID sensor, and the corresponding nodes are activated, accompanied by spreading activation that activates several other nodes in varying degrees. The constructed SN containing context node and all connected nodes are sampled every 500ms and the activation values of each node (except the context node) are stored. This data may be displayed in activation charts as shown with an example in Fig. 2.

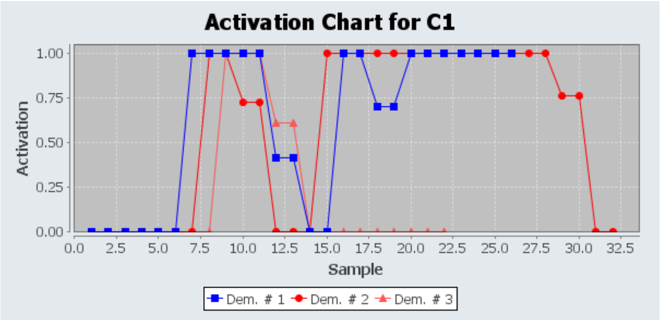

Figure 2. Activation chart of an object denoted "C1". The x-axis is the seconds and the y-axis is the activation values.

Similar activation charts can be constructed also for nodes that received activation through spreading activation.

In the shown example, the total number of samples is 26 for the first demonstration, and object "C1" appeared from the 7th sample. The activation values for samples 0 to 6 are set to 0 and the rest are set through the sensing mechanism (in our case the RFID tag reader) or propagation from other nodes.

At the end of each demonstration, the new context node and its connections can be established. Fig. 3 shows a sample learned context and its connections after the demonstrations. This procedure elaborated in section 5.

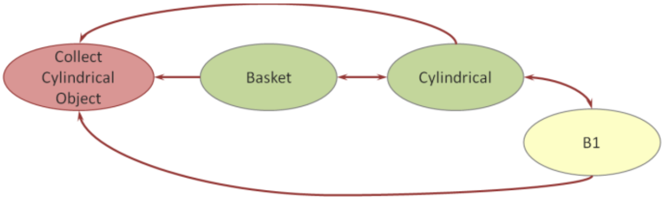

Figure 3. *"Collect Cylindrical Object"* context learned as a result of series of demonstrations.

At the end of each demonstration, the constructed network (e.g. Fig. 3) which is a small portion of predefined SN (Fig. 1) including the context node is used for applying ant algorithms.

4.1.1 Proposed Methodology

The ACO-based learning algorithm is developed according to the general design methodology for Swarm Intelligence tools (Jevtić, 2011). The methodology steps are applied as follows:

- The nodes of the semantic network constitute the discrete data space in which the ants move using as paths the connections between the nodes.
- $\tau(r, s)$ is pheromone update function and $\eta(r, s)$ corresponds to number of connections from the starting node to the context node. Both are used for node transition rule in equation (5).

- Roulette rule is applied as the underlying decision-making mechanism.
- The objective is to keep nodes that their connections propagate higher amount of activation to the context node. Therefore, provided solutions are evaluated by a threshold to identify which nodes in the built path must be remained or removed.

AS and ACS algorithms are applied with the aim to identify and strengthen connections which can fire more activation from the activated node to the context node. Therefore, we are interested more in nodes with less distance to the context node. As an example, according to Fig. 3, *"B1"* connects directly (one level away) to *"Collect Cylindrical Object"*, while *"B1"* is 2 levels away from *"Basket"* to *"Collect Cylindrical Object"*. As a result, *"B1"* to *"Collect Cylindrical Object"* connection is more attractive to the ants. In this strategy, nodes closer to the context node are considered more relevant; therefore their connections will receive higher weight values.

An important point to emphasize is the way in which the ant algorithms are used in this paper. We are not only searching for the shortest path with minimum number of connections to the context node. Rather, we are interested in the amount of pheromone laid on each connection since we can calculate the weight values of the connections based on their pheromone levels. In this paper, weights and pheromone levels are considered the same; the only parameter that determines which nodes are irrelevant is the **pheromone threshold**. By setting a value between 0 and 1, connections below the threshold will be removed and weights are calculated only for the remaining connections. According to equations (4) and (10), the amount of pheromones laid by each ant on a connection depends on two parameters: i) the activation value of the start node ii) the number of levels from the starting node to the context node. Therefore, nodes that receive high activation values and are close to the context node (few levels) are considered more relevant. This is due to the ants that lay more pheromones on these connections, which increase the probability of other ants choosing the same path in subsequent iterations.

The number of ants in each iteration equals the number of samples in each demonstration. For each node in the SN shown in Fig. 3, except the context node, there will be an ant traversing from the node and finishing in the context node. After reaching the context node, the ant is taken out. The procedure is as follows:

1) The number of samples in the demonstration is determined.

2) The list of nodes that are activated and assigned to the context node is provided.

3) For each node in the list and for each sample, an ant will be released to traverse the network.

4) Node transitions and pheromone update are performed according to (5) and (3) for AS and (7) and (9) for ACS.

5) When an ant reaches the context node, it will be taken out.

6) Steps 3 to 5 will be run with fixed number of iterations.

7) Repeat steps 1 to 6 for each demonstration.

8) Pheromone levels of each connection are normalized between 0 and 1.
9) Connections with pheromones above a given threshold remain connected and the rest are removed.
10) Weight values of the remained connections are set to the calculated pheromone levels.

Regarding generalization, there are two approaches:

a) Before applying ACO algorithm, system controls degree of generalization according to the energy values and spreading activation mechanism. This will result in a learning network based on predefined SN that has only a context node and set of activated nodes. So, ACO algorithms are only used for determining the most relevant connections in the learning network and calculating their respective weight values.

b) The ant algorithms determine the amount of generalization and which connections should be strengthened.

In our experiments we used the first approach.

Since this research does not focus on learning low-level representation of behaviors, we applied *Predictive Sequential Learning (PSL)* algorithm for learning and executing low-level behaviors that are mappings of sensory-motor states (Billing and Hellström, 2010; Billing et al., 2010; Billing et al., 2008). Furthermore, we have developed a cognitive architecture that utilizes hard-coded action-primitive pairs instead of PSL to execute simple low-level behaviors (Fonooni et al., 2013). The high-level learning technique based on ACO algorithms can be integrated with both approaches. Hence, we run the experiments regardless of which approach for low-level learning and control has been selected.

4.2 Reproduction and Behavior Arbitration

In order to reproduce a learned behavior, a context must be activated, which means that the conditions learned for the context must be fulfilled. The robot senses the environment with the RFID reader and perceived RFID tags activate corresponding nodes in the semantic network. Activation levels of nodes propagate according to spreading activation mechanism elaborated in section 2, and may in this process activate one or several learned contexts. The highest activated context will be selected, and the robot executes the associated behavior as described in (Fonooni et al., 2013).

Behavior arbitration refers to process of taking a control from one module of an architecture and delegate it to another module (Scheutz, 2002). In our architecture design (Fonooni et al., 2013), high-level controller is dominant over low-level controller meaning that all actions performed by the robot must initiate from the high-level controller. However, low-level controller has the ability to learn sensory-motor skills and control the robot solely. Therefore, reproduction refers to arbitrating the behavior by performing an actuator action according to the corresponding highly activated context.

5. Experimental Results

In the presented experiments the robot is taught to collect objects with particular shapes, and then place them in designated baskets regardless of their color and size. The shapes are *cylindrical*, *triangular* and *square* and the objects should be placed in baskets with the same shape. We run the experiment with three objects with the same shape such that two of them are used for learning, and one is used for reproduction. Each object has associated color, shape and type as depicted in Fig. 1. Collecting and placing in the right basket will be demonstrated separately in two consecutive demonstrations. For the *"Collect Cylindrical Object"* context, the robot recognizes "C1" and "B1" objects ("C1" is a medium sized blue cylindrical object and "B1" is a blue cylindrical basket) by their RFID tags. As a result, the *"C1"* and *"B1"* nodes in the predefined SN are activated and spread activation to all connected nodes. The tutor teleoperates the robot to get close enough to "C1", grabs it and places it in the basket. Finally, the tutor signals the system that learning of the context is completed. A similar demonstration is run a second time with the "C2" and "B1" objects. "C2" is a small sized red cylindrical object. Activation charts of perceived nodes in both demonstrations are shown in Fig. 4.

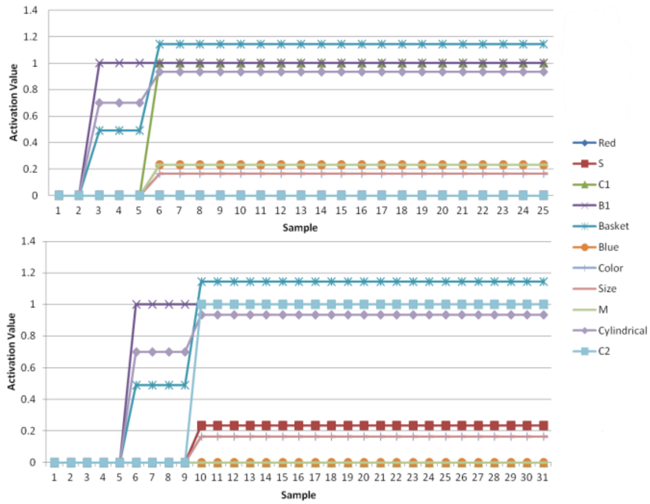

Figure 4. Node activation levels for learning the *"Collect Cylindrical Object"* context perceived in two demonstrations.

Subsequently, learning of the *"Collect Triangular Object"* context begins with teleoperating the robot to grab "T1" and place it in "B2". "T1" is an extra small sized blue triangular object and "B2" is a red triangular basket. "T2", a small red triangular object is used in the second demonstration along with "B2" as the basket. Activation charts of perceived nodes in both demonstrations are shown in Fig. 5.

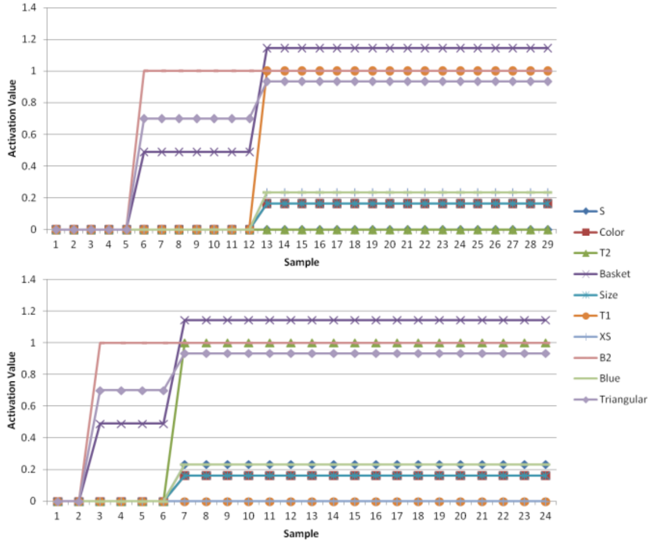

Figure 5. Node activation levels for learning the *"Collect Triangular Object"* context perceived in two demonstrations.

Finally, learning of the *"Collect Square Object"* context is carried out with "SQ1", a large sized green square object and "B3" which is a square shaped basket. In the second demonstration, "SQ2", a medium sized blue square object is used together with "B3". Fig. 6 shows the activation charts of perceived nodes in both demonstrations.

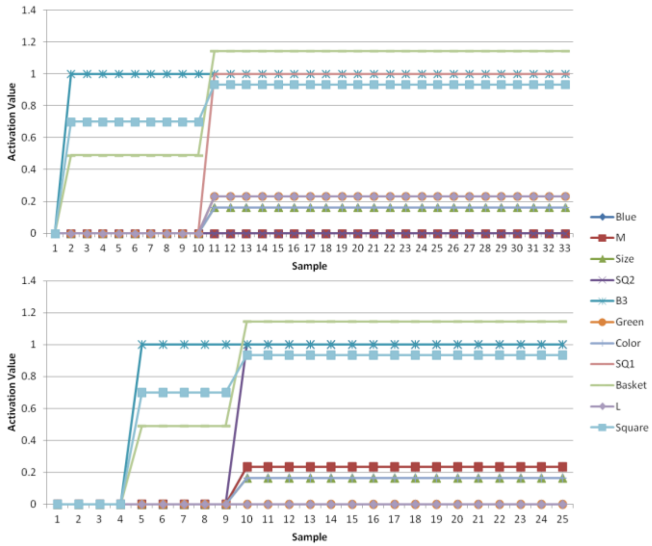

Figure 6. Node activation levels for learning the *"Collect Square Object"* context perceived in two demonstrations.

AS and ACS are then run on the collected data to identify which nodes to connect to each context. These connections will be trimmed and the weight values will be calculated for the remaining connections.

5.1 AS Results

In the first experiment we used AS algorithm with following parameters:

$\rho = 0.1$, $\beta = 2$, pheromone threshold = 0.7, iteration = 30.

Given a predefined SN according to Fig. 1, energy threshold equals 0.2 and decaying factor equals 1.0, system generalizes to the whole network. Table I lists normalized pheromone values for connections between each node and the *"Collect Cylindrical Object"* context node which are also used as weight values.

TABLE I. PHEROMONE VALUES FOR CONNECTIONS TO THE "COLLECT CYLINDRICAL OBJECT" CONTEXT. THE CALCULATED VALUES ARE BASED ON THE AS ALGORITHM.

Node	Pheromone Value / Weight
B1	1.0
Basket	0.9834
Cylindrical	0.8006
C2	0.6889
C1	0.5984
Blue	0.1433
M	0.1239
S	0.0948
Red	0.0761
Size	0.0192
Color	0.0

According to our learning goal, the robot should be able to collect objects and place them in the corresponding basket regardless of their color and size. The results show that the AS algorithm is capable of distinguishing between relevant and irrelevant nodes by giving highest pheromone level to the *"B1", "Basket"* and *"Cylindrical"* nodes. This means the context is activated by **any cylindrical object** together with **"B1"**, the basket designated for cylindrical objects. Fig. 7 shows the final *"Collect Cylindrical Object"* context formed by AS algorithm. The green dashed lines are established during the learning phase and weights are calculated only for these links. The red lines are showing associations which are provided in the pre-defined SN.

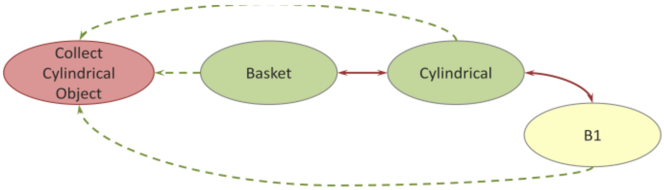

Figure 7. *"Collect Cylindrical Object"* context learned with AS algorithm after two demonstrations.

Table II lists pheromone values of connections between each node and the *"Collect Triangular Object"* context node. The same scenario as cylindrical objects has been arranged with the slight change in the type of objects to collect.

TABLE II. PHEROMONE VALUES FOR CONNECTIONS TO THE "COLLECT TRIANGULAR OBJECT" CONTEXT. THE CALCULATED VALUES ARE BASED ON THE AS ALGORITHM.

Node	Pheromone Value / Weight
Basket	1.0
B2	0.8826
Triangular	0.7563
T2	0.5954
T1	0.4654
XS	0.0989
S	0.0337
Blue	0.033
Size	0.0025
Color	0.0

Fig. 8 shows the final *"Collect Triangular Object"* context formed by AS algorithm.

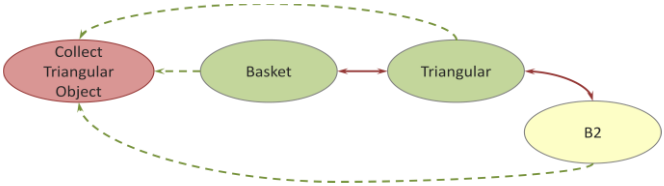

Figure 8. *"Collect Triangular Object"* context learned with AS algorithm after two demonstrations.

In the same way as for cylindrical objects, the AS algorithm identifies the irrelevant nodes. Conditions to be fulfilled in order to activate the context are **any triangular object** together with **"B2"**.

The last context is *"Collect Square Object"*, for which the results are listed in Table III and Figure 9.

TABLE III. PHEROMONE VALUES FOR CONNECTIONS TO THE "COLLECT SQUARE OBJECT" CONTEXT. THE CALCULATED VALUES ARE BASED ON THE AS ALGORITHM.

Node	Pheromone Value / Weight
B3	1.0
Basket	0.9793
Square	0.7583
SQ1	0.6425
SQ2	0.5924
Green	0.1566
M	0.1127
L	0.099
Blue	0.0649
Size	0.002
Color	0.0

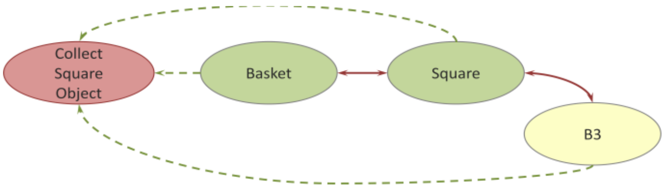

Figure 9. *"Collect Square Object"* context learned with AS algorithm after two demonstrations.

5.2 ACS Results

In this experiment we used the same provided data and run ACS with following parameters:

$\tau_0 = 0.1$, $\rho = 0.1$, $\beta = 2$, $q_0 = 0.9$, threshold=0.7, iteration=30.

The energy threshold and decay factor are set as in for AS, so the system generalizes to the same level. Table IV lists the results of the trimming process for *"Collect Cylindrical Object"* context.

TABLE IV. PHEROMONE VALUES FOR CONNECTIONS TO THE "COLLECT CYLINDRICAL OBJECT" CONTEXT. THE CALCULATED VALUES ARE BASED ON THE ACS ALGORITHM.

Node	**Pheromone Value / Weight**
Basket	1.0
B1	0.8869
C2	0.8804
C1	0.872
Cylindrical	0.7784
Red	0.2495
Blue	0.2484
M	0.2412
Size	0.1917
Color	0.1346

Node	Pheromone Value / Weight
S	0.0

Fig. 10 depicts the final *"Collect Cylindrical Object"* context formed by ACS algorithm.

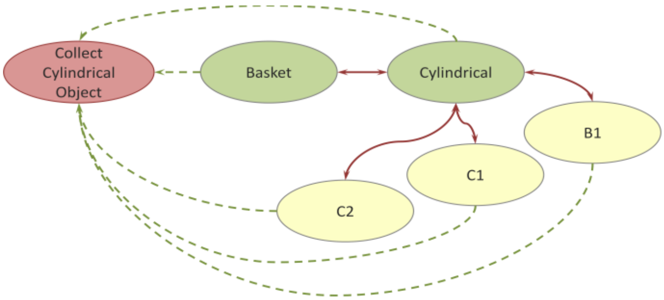

Figure 10. *"Collect Cylindrical Object"* context learned with ACS algorithm after two demonstrations.

While learning the context resulted in almost the same network topology as the one provided by the AS algorithm, ACS identified *"C1"* and *"C2"* nodes as relevant which are not considered relevant by AS. The rationale behind this is that ACS strengthens connections by giving ants that were able to construct the best solution from the beginning of the trial the opportunity to update the pheromone values. Conforming to ACS formulations, nodes with fewer intermediate connections to the context node are considered more relevant and since both *"C1"* and *"C2"* nodes are directly connected to the context node, ants were choosing these paths more often as the others.

Table V lists results of pheromone values of connections between each node and the *"Collect Triangular Object"* context node.

TABLE V. PHEROMONE VALUES FOR CONNECTIONS TO THE "COLLECT TRIANGULAR OBJECT" CONTEXT. THE CALCULATED VALUES ARE BASED ON THE ACS ALGORITHM.

Node	Pheromone Value / Weight
Basket	1.0
B2	0.8893
T1	0.8815
T2	0.8785

Node	Pheromone Value / Weight
Triangular	0.8342
XS	0.269
Size	0.2275
Color	0.2273
Blue	0.2238
S	0.0

Fig. 11 shows the final *"Collect Triangular Object"* context formed by ACS algorithm.

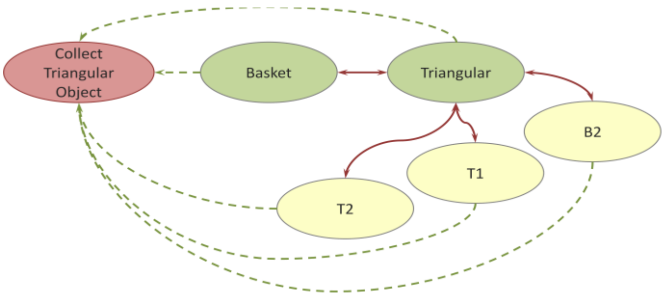

Figure 11. *"Collect Triangular Object"* context learned with ACS algorithm after two demonstrations.

Similarly, *"T1"* and *"T2"* nodes remain connected as well as the other nodes identified previously by the AS algorithm.

Finally, for the last context, *"Collect Square Object"*, the results are listed in Table VI.

TABLE VI. PHEROMONE VALUES FOR CONNECTIONS TO THE "COLLECT SQUARE OBJECT" CONTEXT. THE CALCULATED VALUES ARE BASED ON THE ACS ALGORITHM.

Node	Pheromone Value / Weight
Basket	1.0
B3	0.8725
SQ1	0.8483

Node	Pheromone Value / Weight
Square	0.8109
Green	0.1893
SQ2	0.1818
Blue	0.1805
L	0.1543
Color	0.1268
Size	0.1233
M	0.0

Fig. 12 shows the final *"Collect Square Object"* context formed by ACS algorithm.

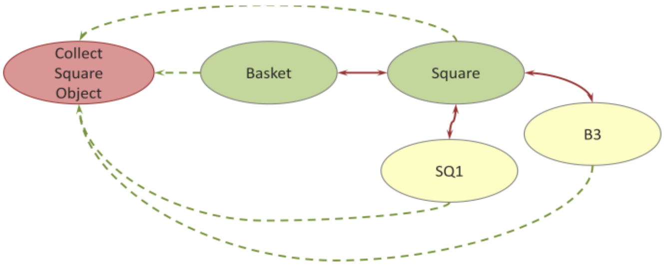

Figure 12. *"Collect Square Object"* context learned with ACS algorithm after two demonstrations.

Unlike the two previous contexts learned by ACS, the *"SQ2"* node is not considered relevant. This will not generate any problems during the reproduction phase since the essential nodes are still determined as relevant.

5.3 Reproduction Results

In order to evaluate the learned contexts, we put the robot in previously unseen situations, thereby showing the generalization ability of the system. The robot should identify a new object, determine its shape and other properties, and activate the appropriate context to collect the object. Depending on the selected context, a low-level actuator action based on PSL or action-primitives in the low-level controller will be executed. Since both AS and ACS algorithms resulted in almost the same network topology, we used contexts learned by AS for our experiments, and all parameters for spreading activation and initial weight values are set as described in section 2.1 plus the decay factor which is set to 0.7.

The "C3" object (extra small yellow cylindrical object), which was not seen by the robot during the learning phase was placed in the environment along with "B1", "B2" and "B3". Perceptions of objects RFID tags activate corresponding nodes in the predefined SN and spreads to all connected nodes. The spreading activation is controlled by equations (1) and (2), and the amount of activation received by each node is illustrated in Fig. 13.

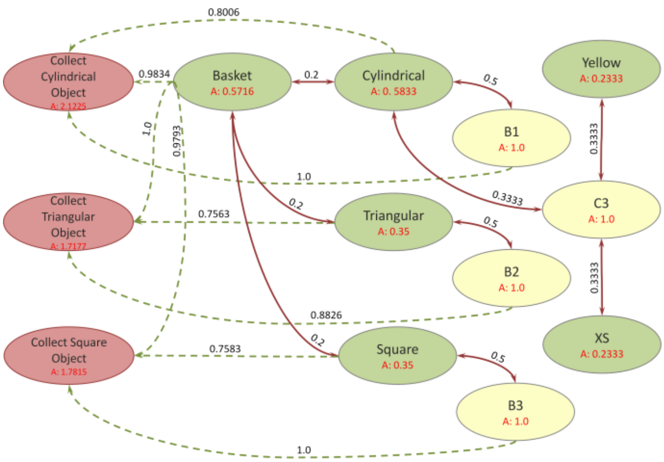

Figure 13. Activation levels of all nodes affected while perceiving "C3", "B1", "B2" and "B3" objects.

As a result, all three contexts are activated to some extent due to the satisfying conditions. The highest activated context is the winner, in this case the *"Collect Cylindrical Object"* with 2.1225 as activation level.

Replacing the "C3" object with "T3" (medium sized yellow triangular object), which is also a new object for the robot, together with "B1", "B2" and "B3", resulted in activation of nodes as shown in Fig. 14.

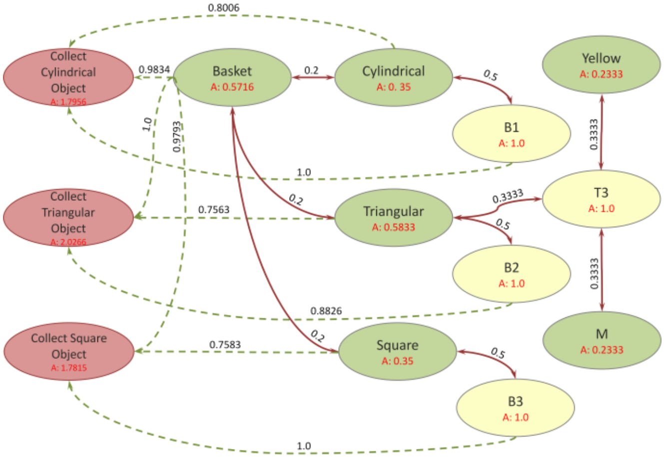

Figure 14. Activation levels of all nodes affected while perceiving “T3”, “B1”, “B2” and “B3” objects.

The highest activated context in this setup is the *“Collect Triangular Object”* with 2.0266 as activation level.

Replacement of “T3” object with a new object “SQ3” (medium sized red square object) in combination with “B1”, “B2” and “B3” resulted in activation of nodes as shown in Fig. 15.

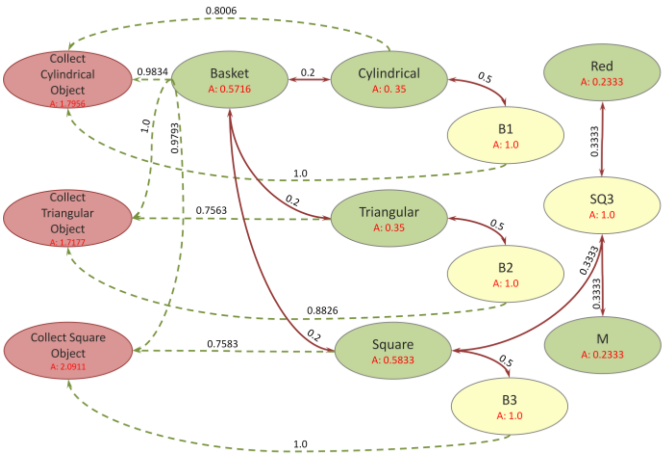

Figure 15. Activation levels of all nodes affected while perceiving “SQ3”, “B1”, “B2” and “B3” objects.

The highest activated context in this arrangement is the *“Collect Square Object”* with 2.0911 as activation level.

The results clearly show that the robot is able to characterize each one of the new objects correctly and engage the relevant contexts accordingly.

The context selection also initiates the process of delivering the selected context to the low-level controller and thus its corresponding action is exhibited by the robot.

6. Conclusions

This paper addresses the challenge of *“What to imitate?”* and *“When to imitate?”* from higher level perspective. In the real world scenarios, concepts plus objects of attention that surround the robot have severe impact on the learning process. Supposing an ambiguous demonstration encompassing numerous distracting objects, a technique to draw the robot’s attention to significant aspects of demonstration is needed. Therefore, a learning method with ACO algorithms and semantic networks for facing the challenges is introduced. The method also employed spreading activation mechanism to provide generalization of concepts while learning high-level representation of behaviors.

As mentioned earlier, the method proposed previously (Fonooni et al., 2013), utilized one-way ANOVA test as a mechanism for identifying relevant elements of demonstrations. This method by its nature is not noise tolerant which makes it inapplicable to scenarios with significant amount of noise. Substitution of ACO algorithms for one-way ANOVA test allows the system to successfully recognize the intentions of the tutor and associate the learned behavior (context) with the correct set of nodes activated through robot's perceptions. Although, the AS and ACS have slight differences in the produced network topologies of the contexts, both are proved to be suitable in learning behaviors. Evaluation of these algorithms is done by the tutor or the end user who clearly aware of the purpose of demonstrations. Therefore they can check whether the robot has learned the associations correctly or the demonstration must be repeated again. In case of learning a behavior correctly, robot’s actions must have the same effects to the world as the one demonstrated by the tutor and thus have to achieve the same goals. Currently, quantitative evaluations are not possible, since there are no other similar methods that can model behaviors in the same manner to test the scenarios.

The high-level learning method is not sufficient for completely reproduce learned behaviors. Hence, there has to be a sophisticated infrastructure such that it can manage both high and low level representations of behaviors. Such an architecture has been proposed in (Fonooni et al., 2013) and tested under a complex scenario for behavior arbitration.

There are several concerns for the future work to take into consideration: i) Defining inhibitory links to represent absence of objects such that the robot may learn skills that require non-presence of objects and/or concepts. ii) Investigating the applicability of priming effects (Neely, 1991) as a bias in determining relevancy of observed objects and concepts in demonstrations. The priming effect can act as a pre-activation method to strengthen desired aspect of demonstration, and to help the learning algorithm to make correct adjustments to the learned associations.

Acknowledgement

This work was financed by the EU funded Initial Training Network (ITN) in the Marie-Curie People Programme (FP7): INTRO (INTeractive RObotics research network), grant agreement no.: 238486.

References

Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469–483.

Bensch, S. & Hellström, T. (2010). On ambiguity in robot learning from demonstration. (pp. 47–56).

Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Handbook of robotics chapter 59: Robot programming by demonstration.

Billing, E. A. & Hellström, T. (2008). Behavior recognition for segmentation of demonstrated tasks. In *Distributed Human Machine Systems, 2008 IEEE SMC International Conference on* (pp. 228–234).: IEEE.

Billing, E. A. & Hellström, T. (2010). A formalism for learning from demonstration. *Paladyn Journal of Behavioral Robotics*, 1(1), 1–13.

Billing, E. A., Hellström, T., & Janlert, L.-E. (2010). Behavior recognition for learning from demonstration. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (pp. 866–872).: IEEE.

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, *2*(4), 353-373.

Breazeal, C. & Scassellati, B. (2002b). Robots that imitate humans. *Trends in Cognitive Sciences*, 6(11), 481–487.

Cakmak, M., DePalma, N., Arriaga, R., & Thomaz, A. L. (2009). Computational benefits of social learning mechanisms: Stimulus enhancement and emulation. In *Development and Learning, 2009. ICDL 2009. IEEE 8th International Conference on* (pp. 1–7).: IEEE.

Chao, C., Cakmak, M., & Thomaz, A. L. (2011). Towards grounding concepts for transfer in goal learning from demonstration. In *Development and Learning (ICDL), 2011 IEEE International Conference on*, volume 2 (pp. 1–6).: IEEE.

Coradeschi, S., & Saffiotti, A. (2003). An introduction to the anchoring problem. *Robotics and Autonomous Systems*, *43*(2), 85-96.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6), 453–482.

Dautenhahn, K. & Nehaniv, C. L. (2002). *The agent-based perspective on imitation*. MIT Press.

Demiris, J. & Hayes, G. (2002). Imitation as a dual-route process featuring predictive and learning components; biologically plausible computational model. *Imitation in animals and artifacts*, (pp. 327–361).

Deneubourg, Aron, S., Goss, S., and Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168.

Dillmann, R. (2004). Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2), 109–116.

Dorigo, M. & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1), 53–66.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *26*(1), 29-41.

Dorigo, M., & Stützle, T. (2003). The ant colony optimization metaheuristic: Algorithms, applications, and advances. In *Handbook of metaheuristics* (pp. 250-285). Springer US.

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. MIT press.

Ekvall, S. (2007). *Robot task learning from human demonstration*. (Doctoral dissertation, KTH Stockholm).

Ekvall, S. & Kragic, D. (2005). Grasp recognition for programming by demonstration. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (pp. 748–753).: IEEE.

Erlhagen, W., Mukovskiy, A., Bicho, E., Panin, G., Kiss, C., Knoll, A., Van Schie, H., & Bekkering, H. (2006). Goal-directed imitation for robots: A bio-inspired approach to action understanding and skill learning. *Robotics and autonomous systems*, 54(5), 353–360.

Fonooni, B., Hellström, T., & Janlert, L.-E. (2012). Learning high-level behaviors from demonstration through semantic networks. In *Proceedings of 4th International Conference on Agents and Artificial Intelligence* (pp. 419–426).

Fonooni, B., Hellström, T., & Janlert, L.-E. (2013a). Applying ant colony system algorithm for high-level behavior learning and reproduction from demonstrations.

Galindo, C., Fernández-Madrigal, J. A., González, J., & Saffiotti, A. (2007, April). Using semantic information for improving efficiency of robot task planning. In *ICRA Workshop: Semantic Information in Robotics*.

Hajimirsadeghi, H., Ahmadabadi, M. N., Araabi, B. N., & Moradi, H. (2012). Conceptual imitation learning in a human-robot interaction paradigm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2), 28.

Huang, Q., Su, J., Zeng, Y., & Wang, Y. (2006). Spreading activation model for connectivity based clustering. In *Advances in Information Systems* (pp. 398-407). Springer Berlin Heidelberg.

Jevtić, A. (2011). *Swarm intelligence: novel tools for optimization, feature extraction, and multi-agent system modeling* (Doctoral dissertation, Telecomunicacion).

Mahmoodian, M., Moradi, H., Ahmadabadi, M. N., & Araabi, B. N. (2013). Hierarchical concept learning based on functional similarity of actions. In *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on* (pp. 1–6).: IEEE.

Marsh, L., & Onof, C. (2008). Stigmergic epistemology, stigmergic cognition. *Cognitive Systems Research*, *9*(1), 136-149.

Mataric, M. J. (2002). Sensoryf: Linking perception to action and biology to robotics. In *Imitation in Animals and Artifacts* (pp. 391–422).: MIT Press.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine learning*, 1(1), 47–80.

Neely, J. H. (1991). Semantic priming effects in visual word recognition: A selective review of current findings and theories. *Basic processes in reading: Visual word recognition*, 11, 264–336.

Pastor, P., Hoffmann, H., Asfour, T., & Schaal, S. (2009). Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on* (pp. 763–768).: IEEE.

Rogers, T. (2008). Computational models of semantic memory. *The Cambridge handbook of computational psychology*, (pp. 226–266).

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6), 233–242.

Scheutz, M. (2002). Affective action selection and behavior arbitration for autonomous robots. In *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, volume 1 (pp. 334–340).

Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden
www.cs.umu.se

ISBN: 978-91-7459-712-7
ISSN: 0348-0542
UMINF: 13.20