University of Ljubljana
Faculty of Computer and Information Science

Alen Vrečko

Merging Multi-Modal Information and Cross-Modal Learning in Artificial Cognitive Systems

MASTER THESIS

Supervisor: assoc. prof. dr. Danijel Skočaj

Ljubljana, 2016

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Alen Vrečko

Združevanje večmodalne informacije in čezmodalno učenje v umetnih spoznavnih sistemih

MAGISTRSKO DELO

Mentor: izr. prof. dr. Danijel Skočaj

Ljubljana, 2016

Izjava o avtorstvu zaključnega dela

Spodaj podpisani Alen Vrečko, vpisna številka 24011292, avtor zaključnega dela z naslovom:

Združevanje večmodalne informacije in čezmodalno učenje v umetnih spoznavnih sistemih (angl. *Merging Multi-Modal Information and Cross-Modal Learning in Artificial Cognitive Systems*)

IZJAVLJAM

1. da sem pisno zaključno delo študija izdelal samostojno pod mentorstvom izr. prof. dr. Danijela Skočaja;
2. da je tiskana oblika pisnega zaključnega dela študija istovetna elektronski obliki pisnega zaključnega dela študija;
3. da sem pridobil/-a vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v pisnem zaključnem delu študija jasno označil/-a;
4. da sem pri pripravi pisnega zaključnega dela študija ravnal/-a v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil/-a soglasje etične komisije;
5. soglašam, da se elektronska oblika pisnega zaključnega dela študija uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
6. da na UL neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja pisnega zaključnega dela študija na voljo javnosti na svetovnem spletu preko Repozitorija UL;
7. dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija.

V Ljubljani, dne 27. avgusta 2016 Podpis študenta/-ke:

Zahvala gre najprej izr. prof. dr. Danijelu Skočaju za mentorstvo pri izdelavi te naloge, za uspešno preteklo sodelovanje, katerega sadovi so vidni tudi v pričujočem delu, in nenazadnje tudi za odločilno vzpodbudo ter vsestransko pomoč pri zaključevanju magistrskega študija.

Za usmerjanje pri študiju in raziskovanju, ki se prav tako odraža v tem delu, se zahvaljujem študijskemu mentorju prof. dr. Alešu Leonardisu.

Zahvala gre članom Laboratorija za umetne vizualne spoznavne sisteme za pomoč in podporo pri zaključevanju študija (ter za skrbno negovanje tradicije dobre jutranje kave).

Za potrpežljivost, skrb in podporo se zahvaljujem sodelavcem na Nacionalnem inštitutu za javno zdravje.

Velika zahvala za potrpežljivost in neomajno podporo gre mami Aniti.

Finally, I would like to express my gratitude to all my colleagues from CogX for their invaluable contributions to this project, and for making George so curious, and Dora so bold.

Contents

List of used acronyms

acronym	english	slovensko
CAST	CoSy Architecture Schema Toolkit	orodje Cosyjeve arhitekturne sheme
DOF	Degrees-of-freedom	prostostne stopnje
HSL	Hue-saturation-luminosity	odtenek-nasičenost-svetilnost
MCMC	Markov chain Monte Carlo	markovska veriga Monte Carlo
MLN	Markov logic networks	markovska logična omrežja
MN	Markov network	markovska mreža
odKDE	On-line discriminative kernel density estimator	diskriminativno sprotno ocenjevanje porazdelitev z jedri
PDF	Probability density function	funkcija gostote verjetnosti
RANSAC	Random sample consensus	konsenz z naključnim vzorčenjem
RGB	Red-green-blue	rdeča-zelena-modra
RGB-D	Red-green-blue-depth	rdeča-zelena-modra-globina
SA	Sub-architecture	podarhitektura, podsistem
SOI	Space of interest	območje zanimanja
TTS	Text-to-speech	iz besedila v govor
WM	Working memory	delovni pomnilnik

Povzetek

Naslov: Združevanje večmodalne informacije in čezmodalno učenje v umetnih spoznavnih sistemih

Čezmodalno povezovanje je združevanje dveh ali več modalnih predstavitev lastnosti neke entitete v skupno predstavitev. Gre za eno temeljnih lastnosti spoznavnih sistemov, ki delujejo v kompleksnem okolju. Da bi se spoznavni sistemi uspešno prilagajali spremembam v dinamičnem okolju, je potrebno mehanizem čezmodalnega povezovanja nadgraditi s čezmodalnim učenjem. Morebiti še najtežja naloga pa je integracija obeh mehanizmov v spoznavni sistem. Njuna vloga v takem sistemu je dvojna: premoščanje semantičnih vrzeli med modalnostmi ter mediacija med nižjenivojskimi mehanizmi za obelavo senzorskih podatkov in višjenivojskimi spoznavnimi procesi, kot sta npr. motivacija in načrtovanje.

V magistrski nalogi predstavljamo pristop k modeliranju verjetnostnega večmodalnega združevanja informacij v spoznavnih sistemih. S pomočjo markovskih logičnih omrežij formuliramo model čezmodalnega povezovanja in učenja ter opišemo načela njegovega vključevanja v spoznavne arhitekture. Prototip modela smo ovrednotili samostojno, z eksperimenti, ki simulirajo trimodalno spoznavno arhitekturo. Na podlagi našega pristopa oblikujemo, implementiramo in integriramo tudi podsistem prepričanj, ki premošča semantični prepad v prototipu spoznavnega sistema George. George je inteligenten robot, ki je sposoben zaznavanja in prepoznavanja predmetov iz okolice ter učenja njihovih lastnosti s pomočjo pogovora s človekom. Njegov poglavitni namen je preizkus različnih paradigem o interaktivnemu učenju konceptov. V ta namen smo izdelali in izvedli interaktivne eksperimente za vrednotenje Georgevih vedenjskih mehanizmov. S temi eksperimenti smo naš pristop k večmodalnemu združevanju informacij preizkusili in ovrednotili tudi kot del delujočega spoznavnega sistema.

Ključne besede: čezmodalno povezovanje, čezmodalno učenje, spoznavni sistemi, iteligentni roboti, markovska logična omrežja, strojno učenje, umetna inteligenca.

Abstract

Title: Merging Multi-Modal Information and Cross-Modal Learning in Artificial Cognitive Systems

Cross-modal binding is the ability to merge two or more modal representations of the same entity into a single shared representation. This ability is one of the fundamental properties of any cognitive system operating in a complex environment. In order to adapt successfully to changes in a dynamic environment the binding mechanism has to be supplemented with cross-modal learning. But perhaps the most difficult task is the integration of both mechanisms into a cognitive system. Their role in such a system is two-fold: to bridge the semantic gap between modalities, and to mediate between the lower-level mechanisms for processing the sensory data, and the higher-level cognitive processes, such as motivation and planning.

In this master thesis, we present an approach to probabilistic merging of multi-modal information in cognitive systems. By this approach, we formulate a model of binding and cross-modal learning in Markov logic networks, and describe the principles of its integration into a cognitive architecture. We implement a prototype of the model and evaluate it with off-line experiments that simulate a cognitive architecture with three modalities. Based on our approach, we design, implement and integrate the belief layer – a subsystem that bridges the semantic gap in a prototype cognitive system named George. George is an intelligent robot that is able to detect and recognise objects in its surroundings, and learn about their properties in a situated dialogue with a human tutor. Its main purpose is to validate various paradigms of interactive learning. To this end, we have developed and performed on-line experiments that evaluate the mechanisms of robot's behaviour. With these experiments, we were also able to test and evaluate our approach to merging multi-modal information as part of a functional cognitive system.

Keywords: binding, cross-modal learning, cognitive systems, intelligent robots, Markov logic networks, machine learning, artificial intelligence.

Chapter 1

Introduction

Cognitive systems can be best described as systems able of understanding information in order to make informed decisions. To do that, they have to be capable of performing specific cognitive operations, like analysing, relating, deciding, planning, etc. An artificial cognitive system operating in a real world environment must be able to collect relevant information about its surroundings, understand it, and make autonomous decisions or plans about its activities within the same environment. In general, the information about the environment can be collected in two ways: (i) by interpreting data from sensors, i.e. by *perception*, or (ii) by interpreting information from another agent, if the system is capable of *communication* with him.

Perception is, of course, the more direct and efficient of the two ways. It involves transforming the sensory data into a suitable and usually more general representation that can be used by complex cognitive mechanisms (we say that such representations are grounded in system's sensory input). In this process, the system relies on its conceptual knowledge about the environment. Perception is therefore a cognitive process that merges sensory data from the environment with conceptual information that is part of the system knowledge.

Since a cognitive system can have multiple types of sensors, each of them with its own characteristics and specifics, a very important part of its cognition is the ability to merge representations from multiple sources (in this work we call them *modalities*) into a unified representation (in other words, merging different kinds of perceptions of the same thing into a single notion). A specific type of knowledge that allows the system to relate information from different modalities is needed for such a process. We call such knowledge the *cross-modal knowledge*.

When a cognitive system operates in an ever-changing, dynamic environment, its ability to adapt to changes in such environment becomes vital. This ability translates to various cognitive mechanisms that allow the system to continuously update its knowledge, accommodating new concepts, or just adapting the old ones. All these mechanisms, of course, involve learning, either to improve perceptive abilities of the system, or to increase its ability to associate between different kinds of perceptions, i.e. *cross-modal learning*. However, to be able to learn something, there must be first a learning opportunity. Therefore, a cognitive system should also incorporate mechanisms of motivation and behaviour that actively seek such opportunities (e.g. searching for a peculiar item with rare properties).

The pursuit of knowledge becomes more varied, if we add another agent to the environment (e.g. a human), and make the system able to communicate with him. The system can exploit the dialogue with the agent to supplement its perception of the environment, e.g. to obtain information about an item that its perception alone can not. Such situations also create learning opportunities. The system can also actively strive to improve its conceptual knowledge, e.g. by deliberately engaging in conversation about a certain concept. However, even in such an environment, the ability to improve autonomously its knowledge is still useful to a cognitive system, although, learning is usually more efficient, when a supervisor is involved.

Situated and non-situated dialogue with another agent and related cognitive mechanisms require yet more sophistication from the cognitive layers that merge multi-sourced information. Besides *multi-modal* information, the system must also manage *multi-agent* information. Many aspects of these two problems are very related, especially, if the system has to deal with one agent, only (in this thesis, we will sometimes treat multi-agent information as multi-modal). However, with the increased autonomy and competencies of the cognitive system, widens also the potential knowledge gap between the system and other agents. Thus, managing multi-agent information becomes also an epistemic problem.

1.1 Goals and Methods

The focus of this thesis is on cognitive processes that relate and merge information from different sources, in order to produce unified representations that can be used by higher-level cognition. They represent a crucial part of any cognitive system operating in a realistic environment. Our aim is to define a paradigm about such processes, develop a method based on this paradigm, implement a prototype of the method, and evaluate the prototype both off-line and as part of a real cognitive system (i.e. a robot operating in a real world environment). The paradigm, the method and the implementation must also include mechanisms for adapting and improving the knowledge that is used for merging information.

We assume an open and uncertain environment, where the system has to be always ready to cope with uncertainty in its perceptions, as well as acknowledge new concepts. This implies a probabilistic approach to modelling internal representations, and consequently probabilistic methods for merging multi-modal information and learning. The methods must be capable of probabilistic modelling of conceptual knowledge that is continuously accumulated, and of forming integrated probabilistic representation of the environment that the system is currently aware of (i.e. perceptions).

Our aim is also to develop an approach to integration of our method into a prototype robot. This will allow us to evaluate our paradigm as part of a real cognitive system, and see how it works in conjunction with other cognitive processes that typically make a cognitive system functional (e.g.

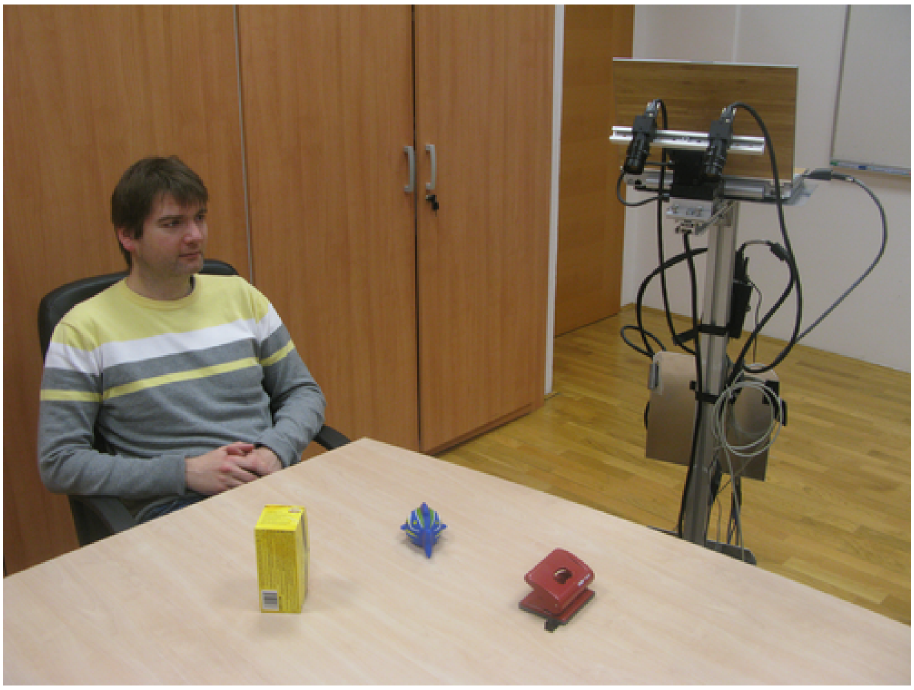

Figure 1.1: George in an early development stage.

situated dialogue, motivation, planning, etc.).

1.2 Contributions

The probabilistic model of binding and cross-modal learning, its formulation in MLN and the approach to its integration into a cognitive system represent the core of this thesis and its main contributions. This work was published in [44].

The robot George, which we use as the evaluation platform for our methods and approaches, is the result of a joint effort of six partners within the FP7 European project *CogX*: University of Birmingham (UK), DFKI (Germany), KTH Stockholm (Sweden), University of Ljubljana (Slovenia), Alfred Ludwig University of Freiburg (Germany) and TU Wien (Austria). The author of this thesis made several contributions to this research. Besides the belief layer and the reference resolution, he contributed to the visual attention mechanism, the attentive part of the visual subsystem (2D object segmentation) and the system integration in general. He also made many other technical contributions, e.g. the robotic arm integration, the pan-tilt unit integration, etc. The description of George that we provide in this thesis is a digest of [39].

Another important contribution of this work are the belief layer and the mechanism for reference resolution in George. They represent the embodiment of our core ideas and methods within a realistic context. The author of this thesis made most of this research in collaboration with Miroslav Janíček from DFKI (Germany) [43].

The mentor of this thesis and the author of this thesis designed together the experiments for evaluation of George's behaviour. The experiments were performed by the author of this thesis.

1.3 Outline of the Thesis

This thesis is organised as follows. In the next chapter, we present the paradigm of *binding* that underlies our approach. We first define the problem that we aim to solve, then we describe our method and its implementation. Chapter 3 is dedicated to George, a prototype of a cognitive system that represents the platform for evaluation of our approaches and methods. In Chapter 4, we describe in more detail the *belief layer* of George. The belief layer is a vital part of George that bridges the *semantic gap* between its lower and higher cognition. The belief layer is the result of our efforts to integrate our approach and methods into George. In Chapter 5, we present two sets of experiments: (i) the off-line experiments on the prototype binding mechanism from Chapter 2, and (ii) the experiments performed on George prototype. Finally, we conclude with Chapter 6, where we summarize this thesis and express our final remarks.

Chapter 2

Binding and Cross-modal Learning

One of the most important abilities of any cognitive system operating in a real world environment is to relate and merge information from different modalities. For example, when hearing a sudden, unexpected sound, humans automatically try to locate visually its source in order to relate the audio perception of the sound to the visual perception of the source. The process of combining two or more modal representations[1] (grounded in different sensory inputs) of the same entity into a single multi-modal representation is called *binding*. While the term *binding* has many different meanings across various scientific fields, a very similar definition comes from neuroscience, where it denotes the ability of the brain to converge perceptual data processed in different brain parts and segregate it into distinct elements [2, 36].

The binding process can operate on different types and levels of cues. In the example above, the direction that the human perceives the sound from is an important cue, but sometimes this is not enough. If there are several

[1] In the literature, the term modality typically refers to a sensory modality, also known as stimulus modality. A modal representation is a collection of information about a physical entity based on a particular sensory input, for example visual, auditory, olfactory, or kinaesthetic information. We adopt here a notion of modality that includes both sensory data, and further interpretations of that data within the modality [38].

potential sound sources in the direction of the percept, the human may have to relate higher-level audio and visual properties. A knowledge base that associates the higher-level perceptual features across different modalities is therefore critical for a successful binding process in any cognitive system.

In order to function properly in a dynamic environment, a cognitive system should also be able to learn and adapt in a continuous, open-ended manner. The ability to update the cross-modal knowledge base on-line, i.e. cross-modal *learning* is therefore vital for any kind of binding process in such an environment.

Many of the past attempts at binding information within cognitive systems were restricted to associating linguistic information to lower level perceptual information. Roy et al. tried to ground the linguistic descriptions of objects and actions in visual and sound perceptions and to generate descriptions of previously unseen scenes based on the accumulated knowledge [34, 35]. This is essentially a symbol grounding problem first defined by Harnad [15]. Chella et al. proposed a three-layered cognitive architecture around the visual system with the middle, *conceptual layer* bridging the gap between linguistic and sub-symbolic (visual) layers [7]. Related problems were also often addressed by Steels [40].

Jacobsson et al. approached the binding problem in a more general way [21, 20] developing a cross-modal binding system that could form associations between multiple modalities and could be part of a wider cognitive architecture. They modelled the cross-modal knowledge as a set of binary functions comparing binding attributes in a pairwise fashion. A cognitive architecture using this system for linguistic reference resolution was presented in [45]. This system was capable of learning visual concepts in interaction with a human tutor. Later, the same group developed a probabilistic binding system that encoded cross-modal knowledge into a Bayesian graphical model [46]. In [27] a framework for constructing higher-level cognitive representations of the environment, called beliefs, was presented. Markov logic was used as the main framework for various types of inference over beliefs, including perceptual grouping, which comes very close to our definition of binding. All these systems assumed static cross-modal knowledge.

Our aim is to develop a flexible binding system, capable to adapt continuously its probabilistic representation of cross-modal knowledge to the challenges of a dynamic environment. These requirements lead us in the direction of Markov graphical models as a powerful and flexible platform for probabilistic problem formulation. Unlike previous binding systems, the system presented here is able to learn cross-modal associations in a continuous manner. As a basis for our work, we also introduce a formal definition of the binding problem, which is still general enough to accommodate other possible approaches to binding.

This chapter is organised as follows: in the next section, we formally define the problems of cross-modal learning and binding. In Section 2.2, we first briefly describe the basics of *Markov logic networks (MLN)*. Then we describe our binding and cross-modal learning model that is based on MLN and, in Section 2.3, discuss its integration in a cognitive architecture.

In order to validate our approach, we dedicate the first part of Chapter 5 (Section 5.1) to the experiments performed on an off-line binding system, designed according to the methods described in this chapter.

2.1 Problem Definition

The main idea of cross-modal learning is to use successful bindings of modal percepts as learning samples for the cross-modal learner. The improved cross-modal knowledge thus enhances the power of the binding process, which is then able to bind together new combinations of percepts, i.e. new learning samples for the learner. For example, if a cognitive system is currently capable of binding an utterance describing something blue and round to a perceived blue ball only by colour association, this particular instance of binding could teach the system to associate also the perceived visual shape of the ball to the linguistic concept of roundness. We see that at least on this level the binding process depends on the ability to associate between *modal features* (in this example the perceived colour and the perceived shape are features of the visual modality, while the linguistic concepts of blueness and roundness belong to the linguistic modality).

We assume an open world in terms of *modal features* (new features can be added, obsolete features retracted). The cross-modal learner starts with just some basic prior knowledge of how to associate between a few basic features, which is then gradually expanded to other features and the new ones that are created.

The cross-modal learning problem is closely related to the problem of *association rule learning* in data mining, which was first defined by Agrawal et al. [1]. Therefore, we will base our problem definition on Agrawal's definition and expand it with the notions of *modalities*, *percepts* and *percept unions.*

We have a set of n binary attributes called *features* $F = \{f_1, f_2, ..., f_n\}$ and a set of rules called the *knowledge database* $K = \{t_1, t_2, ..., t_m\}$. A rule is defined as an implication over two subset of features:

$$t_i : X \Rightarrow Y \tag{2.1}$$

where $X, Y \subseteq F$ and $X \cap Y = \emptyset$. A feature can not be part of both sides of the implication.

The rules can be associated with several additional values – the two most typical are the *feature-set support* and the *rule confidence.* The *feature-set support* is the observed frequency of a combination of features (e.g. $supp(X) = 0.25$). When the feature combination on the left side of a rule is supported by a particular situation (i.e. we have evidence for all the features in the combination), we consider that rule *relevant* for that situation. The *rule confidence* is defined as

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} = P(Y|X). \tag{2.2}$$

It can be interpreted as the estimate of the probability that a relevant rule actually holds.

The features represent various higher-level properties of perceived entities based on the sensory input, while the rules encode associations between those properties relative to a single entity. For example, we could attribute the rule $t : X \Rightarrow Y$ with the confidence $conf(t)$ the following meaning: *all the features from the set X have been perceived in an entity, then we can claim with confidence $conf(t)$ that the same entity has also all the features from the set Y.*

We expand the Agrawal's definition by introducing the notion of *modality.* Modalities are channels of perception based on specific sensors. A modal *feature set* is a collection of features that can be perceived by a single modality. Modal feature sets are thus represented as subsets of the feature set F, where each feature is restricted to one modal subset only:

$$\begin{aligned} &M_1 = \{f_{11}, f_{12}, ..., f_{1n_1}\} \\ &M_2 = \{f_{21}, f_{22}, ..., f_{2n_2}\} \\ &\ldots\ldots \\ &M_k = \{f_{k1}, f_{k2}, ..., f_{kn_k}\} \end{aligned} \tag{2.3}$$

$$\begin{aligned} &M_m \cap M_n = \emptyset, \quad m, n \in \{1, 2, ..., k\},\ m \neq n \\ &F = M_1 \cup M_2 \cup ... \cup M_k \end{aligned}$$

We modify the rule-making restrictions of (2.1) accordingly:

$$\begin{aligned} &1.\ N = M_{m_1} \cup M_{m_2} \cup ... \cup M_{m_r}, \quad m_1, ..., m_r \in \{1, 2, ..., k\},\ r < k \\ &2.\ Y \subseteq N \\ &3.\ X \subseteq F \setminus N \end{aligned} \tag{2.4}$$

We can see that modal sets are restricted to have their member features on one side of the implication, only. This generalization of the original restriction focuses our knowledge base on cross-modal associations. We assume that the intra-modal associations are processed on the lower, modal levels and should not directly influence the processes on the cross-modal level.

Next, we need a means of relating features to perceived entities. *Percepts* are collections of features from a single modality. A percept is the result of intra-modal processing of specific types of sensory signals (usually from one type of sensor or a group of related sensors) that belong to a single entity. A percept acts as uni-modal representation of a perceived entity. Let **P** be the set of current percepts, i.e. the *percept configuration*:

$$\mathbf{P} = \{P_1, P_2, ..., P_n\}, \quad P_i \subseteq M_j. \tag{2.5}$$

In any percept configuration, an entity can be represented with multiple percepts, but not more than one per modality.

Percept unions[2] are collections of percepts from different modalities. A percept union acts as a shared representation of a perceived entity, grounded through its percepts to different types of sensory data. Given the percept configuration **P**, **U**(**P**) is the set of current percept unions, i.e. the *union configuration*:

$$\mathbf{U}(\mathbf{P}) = U_1, U_2, ..., U_m, \quad U_i \subseteq P. \tag{2.6}$$

A percept union U_i can not contain more than one percept per modality. Figure 2.1 illustrates the concepts above with an example.

In this view, the associations between the features in the knowledge database K encode the information about how the percepts bind to percept unions. E.g. let us suppose we have percepts P_1 and P_2 and the rule $t : X \Rightarrow Y$, where $X \subseteq P_1$ and $Y \subseteq P_2$. Then $conf(t)$ can help us estimate how likely it is that P_1 and P_2 belong to the same union. Of course, other measures can be used instead of $conf(t)$ to estimate the plausibility of the rule.

Now we can define the process of binding as a mapping of a percept configuration to one of the possible union configurations:

$$\beta : \mathbf{P} \rightarrow \mathbf{U}(\mathbf{P}), \tag{2.7}$$

[2] Although the term might imply otherwise, we can see in (2.20) that a *percept union* is a set of *percept sets* (a set of sets of features), not a union of *percept sets* (set of features).

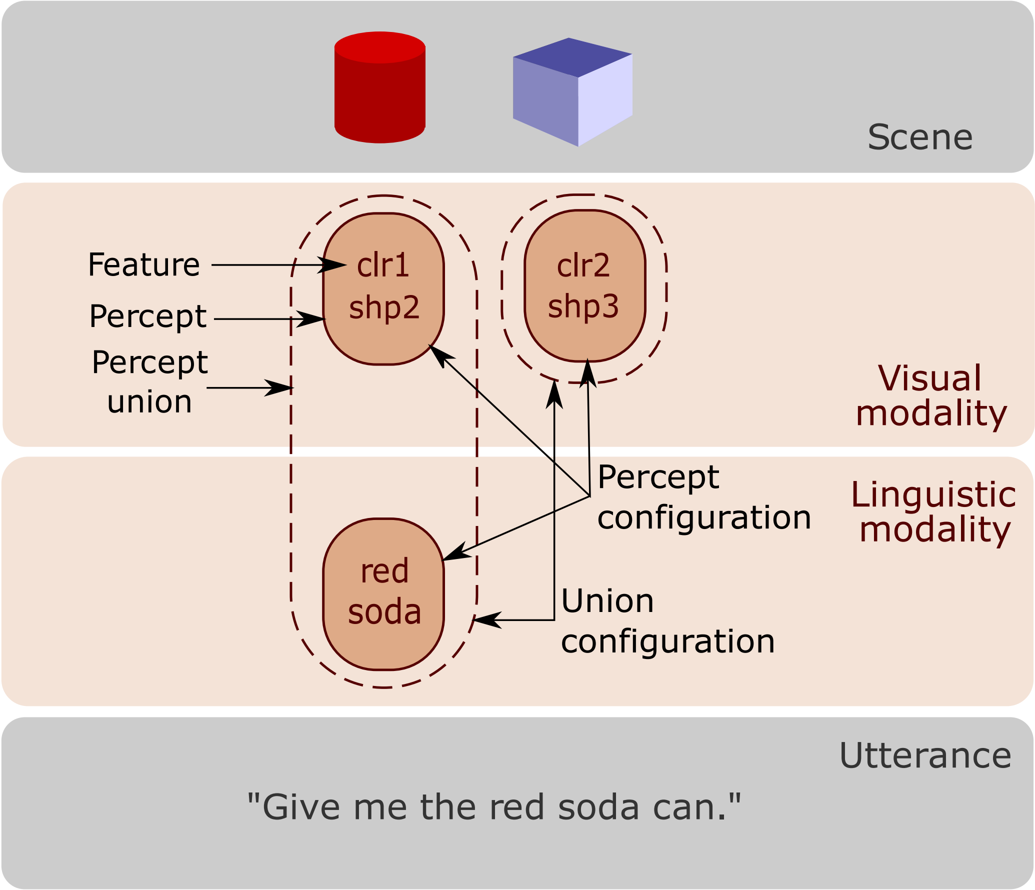

Figure 2.1: A binding example from a human-robot interaction. The robot visual system sees two objects, resulting in two visual *percepts* with *features* for colour and shape. Based on the previously accumulated knowledge, the modal subsystem is able to classify the objects' colours and shapes as 'clr1', 'clr2', 'shp2' and 'shp3'. In general (and especially when modal concepts are learned without human influence), the robot's modal concepts do not necessarily match human perceptions (e.g. the visual system could perceive what we see as red and orange as the same colour). The reference in the sentence uttered by the human results in a percept in the robot's linguistic modality. The ensemble of the three percepts makes the current *percept configuration*. The linguistic percept and the visual percept representing the red cylinder (can) form a single percept union, since they very probably represent the same physical entity, whereas the percept representing the blue box makes a separate union. The ensemble of both unions makes the current *union configuration*.

where the following restrictions apply:

$$\begin{aligned}
&1.\ N = U_1 \cup U_2 \cup ... \cup U_m = \mathbf{P} \\
&2.\ \forall U_i, U_j \in \mathbf{U}(\mathbf{P}),\ i \neq j:\ U_i \cap U_j = \emptyset \\
&3.\ \forall P_i, P_j \in U_k,\ i \neq j:\ P_i \subseteq M_l \wedge P_j \subseteq M_m \Rightarrow l \neq m
\end{aligned} \tag{2.8}$$

The first two binding restrictions assign each percept in the configuration to exactly one union, while the third restricts the maximum number of percepts per modality in a union to one. The third binding restriction follows the assumption from (2.4) that the modal layers are able to produce consistent modal representations of real world entities.

Finally, to make the binding process plausible, we introduce a measure of confidence in a union configuration based on the knowledge K — the *binding confidence* $bconf_K(\mathbf{U}(\mathbf{P}))$. Strict definition of $bconf_K(\mathbf{U}(\mathbf{P}))$ is a matter of implementation of the binding system and depends on measures that we use to estimate the rule plausibility (e.g. $conf(t)$). In general, the system should find for every rule the frequency of support ($supp(X \cup Y)$) and the frequency, with which the rule is violated ($supp(X) - supp(X \cup Y)$). The binding confidence increases or decreases each time a rule is supported or broken according to the rule's plausibility (supported plausible rules and broken implausible rules should increase the confidence, while broken plausible rules and supported implausible rules should decrease it).

Given a percept configuration $\mathbf{P}$ and a knowledge base K, the task of the binding process is to find the optimal union configuration:

$$\mathbf{U_{opt}}(\mathbf{P}) = \underset{\mathbf{U}(\mathbf{P})}{\operatorname{argmax}}(bconf_K(\mathbf{U}(\mathbf{P}))). \tag{2.9}$$

In this sense – i.e. considering $bconf_K(\mathbf{U})$ as a predictor based on K – we can consider higher-level cross-modal learning as a regression problem. Therefore, the aim of the cross-modal learner is to maintain and improve the cross-modal knowledge base, thus providing an increasingly more reliable measure of binding confidence.

2.2 Formulation in Markov Logic Networks

Markov logic networks (MLN) [32, 10, 9] combine first-order logic and probabilistic graphical models in a single representation. An MLN knowledge base consists of a set of first-order logic formulae (rules) with a weight attached:

$$weight \quad first - orderlogicformula. \tag{2.10}$$

The weight is a real number, which determines how strong a constraint each rule is: the higher the weight, the less likely the world is to violate that rule.

Together with a finite set of constants, the MLN defines a *Markov network (MN)* (also called *ground Markov network* or *Markov random field*). A Markov network is an undirected graph where each possible grounding of a predicate (all predicate variables replaced with constants) represents a node, while the formulae define the edges connecting the nodes. Each grounded formula defines a clique in the graph. An MLN can thus be viewed as a template for constructing MNs. In general the probability distribution over possible interpretations x defined by an MN is given by

$$P(X = x) = \frac{1}{Z} \prod \phi_k(x_{\{k\}}), \tag{2.11}$$

where ϕ_k is the potential function of the clique k, $x_{\{k\}}$ is the state of the subset of variables that appear in the clique k, and Z is the partition function defined as

$$Z = \sum_x \prod_k \phi_k(x_{\{k\}}). \tag{2.12}$$

A convenient way to model Markov networks is *logistic regression*, which defines the weight w_t for the formula t as

$$w_t = \ln(\frac{P(t)}{1 - P(t)}), \tag{2.13}$$

where $P(t)$ is the probability that the formula t is not violated. The probability distribution over possible worlds x is then given by

$$P(X = x) = \frac{1}{Z}\exp(\sum_t w_t n_t(x)), \tag{2.14}$$

where $n_t(x)$ is the number of true groundings of the formula t, and

$$Z = \sum_x \exp(\sum_i w_i n_i(x)).$$

The inference in Markov Networks means finding a stationary distribution of the system. Usually we are interested in a marginal distribution of a subset of variables, very often conditioned by some prior knowledge, called *evidence* (another subset of variables whose values are known in advance). Sometimes we are only interested in the most likely state of a subset of variables given some evidence – *Maximum a-posteriori probability estimation (MAP).*

Exact inference in MN is considered a $P\#$-complete problem [33]. Methods for approximating the distribution include various *Markov chain Monte Carlo* sampling algorithms [14] and *belief propagation* [48]. MAP inference in MN represents a *weighted maximum satisfiability problem.*

2.2.1 Cross-modal Knowledge Base

Our cross-modal knowledge base consists of two types of templates for the binding rules. The template for the *aggregative rule* is defined as

$$perFeat(p_1, f_1) \wedge uniPer(u, p_1) \wedge perFeat(p_2, f_2) \Rightarrow uniPer(u, p_2), \tag{2.15}$$

where the predicate $perFeat(p, f)$ denotes that the feature f is part of the percept p, while $uniPer(u, p)$ denotes that union u includes the percept p. Variables (p_1, f_2, u, etc.) begin with a lowercase character. In a very similar manner we define the template for the *segregative rule*:

$$perFeat(p_1, f_1) \wedge uniPer(u, p_1) \wedge perFeat(p_2, f_2) \Rightarrow \neg uniPer(u, p_2). \quad (2.16)$$

We can identify the aggregative rules as the mechanism that merges percepts into common percept unions, while segregative rules separate them in distinct unions. The template rules represent a subset of associative rules in (2.1), restricted with (2.4), where each side is limited to one feature.

We also define the binding domain that we will use to ground the network. A domain is a collection of typed constants. The following is an example of binding domain with two modalities:

$$\begin{aligned} modality &= \{Language,\ Vision\} \\ feature &= \{Red,\ Green,\ Blue,\ Compact,\ Flat,\ Elongated, \\ &\quad Box,\ Ball,\ Soda, \\ &\quad Clr1,\ Clr2,\ Clr3,\ Shp1,\ Shp2,\ Shp3\}. \end{aligned} \quad (2.17)$$

We can see that the constants (beginning with an uppercase character) can be of two types: *modalities* and *features*. The predicate $modPart(mod,\ feat)$ is used to determine the partition of the features between modalities in the sense of (2.3). For example:

$$\begin{aligned} &modPart\{Language,\ Red\}, \\ &modPart\{Vision,\ Clr1\}. \end{aligned}$$

Based on the example domain (2.17) a small set of grounded and weighted binding rules could look like this:

$$\begin{aligned} &2.5 \quad perFeat(p_1, Red) \wedge uniPer(u, p_1) \wedge perFeat(p_2, Clr1) \Rightarrow uniPer(u, p_2) \\ &1.9 \quad perFeat(p_1, Red) \wedge uniPer(u, p_1) \wedge perFeat(p_2, Clr2) \Rightarrow \neg uniPer(u, p_2). \end{aligned} \quad (2.18)$$

At this stage, the predicates forming the binding rules are not fully grounded yet. They are grounded on the conceptual level only, with known features like *Red*, $Clr1$, etc., while the unions are still represented with variables.

In general the rules are fully grounded each time an inference is performed, when based on some perceptual information (e.g. objects that are currently perceived by a robot) an MN is constructed. We call the former process *concept grounding* and the latter process *instance grounding*. Such approach to grounding, i.e. *staged grounding*, can be very beneficial for a cognitive system. While decoupling the general from the specific, it allows for the application and adaptation of general concepts learned over longer periods of time to the current situation in a very flexible fashion.

Using the example domain in (2.17), we can formulate the percept configuration in figure 2.1 like

$$\begin{aligned}&perFeat(P1, Clr1) \wedge perFeat(P1, Shp2)\ \wedge\\&perFeat(P2, Clr2) \wedge perFeat(P2, Shp3)\ \wedge\\&perFeat(P3, Red) \wedge perFeat(P3, Soda).\end{aligned} \tag{2.19}$$

We ground the possible percept unions with constants $\{U1, U2, ...\}$. From (2.18) and (2.19) we can infer the following union configuration as the most probable:

$$uniPer(U1,\ P1) \wedge uniPer(U1,\ P3) \wedge uniPer(U2,\ P2). \tag{2.20}$$

Percepts $P1$ and $P3$ are bound to a common percept union $U1$, while the percept $P2$ is part of a separate percept union $U2$.

Besides the binding rules, the database can also contain *feature priors* in the following form:

$$weight \quad perFeat(percept,\ feature).$$

A feature prior denotes the default probability of a feature belonging to a percept, which is used if there is no positive or negative evidence about it. The feature priors can be based on the past observations, e.g. making the red colour more likely than the pink because observed more often. Alternatively, we can decide for arbitrary priors, e.g. uniformly distributed within a certain feature type, regardless of the observations (in this case, the probability to observe a colour is equal for all known colours).

In a similar fashion to the predicate *modPart*, we can use the predicate *typePart* to further discriminate between the feature types within modalities. For example, if we want to distinguish between colours and shapes within the visual modality (this is particularly useful in the case of arbitrary priors) or maybe restrict modal percepts to just one feature from a particular group. Such partitions are treated as intra-modal processes, therefore they should be provided by the modal subsystems.

2.2.2 Learning

After the rules and priors (if we did not opt for arbitrary priors) are grounded within the binding domain, we need to learn their weights. We use the generative learning method described in [32]. The learner computes a gradient from the weights based on the number of true groundings ($n_i(x)$) in the learning database and the expected true groundings according to the MLN ($E_w[n_i(x)]$):

$$\frac{\delta}{\delta w_i} \log P_w(x) = n_i(x) - E_w[n_i(x)], \tag{2.21}$$

and optimizes the weights accordingly. Since the expectations $E_w[n_i(x)]$ are very hard to compute, the method uses the *pseudo-likelihood* to approximate it [3].

Continuous learning is performed by feeding the percept unions to the system in small batches. In this way, we make sure that the learning affects both aggregative and segregative rules in equal measure. Each small batch thus represents one learning sample and typically contains 2-5 percept unions described with *perFeat* and *uniPer* predicates.

In each learning step, the learner accepts the rule's old weight in the knowledge database as the mean for the Gaussian prior, which it tries to adjust based on the new training batch. By setting the dispersion of the weight's Gaussian prior to an adequate value, we ensure the learning rate of each learning sample is proportional to its size.

In an on-line scenario, successfully inferred union configurations should also be used as learning samples. In this case, the logistic model insures that among the rules supported by the learning sample, the learning impact is more pronounced on those with smaller absolute weight values. This allows the system to increase the knowledge about new cross-modal concepts without overfitting the old ones.

2.2.3 The Binding Process

In MLN, the binding process from (2.9) translates to inferring the value of certain predicates from a graphical model (MN) over some evidence. As already seen in Section 2.2.1, an MN is a result of the processes of concept and instance grounding. Typically, we are querying for values of the predicates *uniPer*, where evidence includes a description of the current percept configuration (using the predicate *perFeat*), a list of known and potential percept unions, and a description of the current partial union configuration (percepts can be assigned to already known unions). In the case of probabilistic inference (e.g. MC-SAT, Gibbs sampling, etc.) the binding result is expressed as a probability distribution for each unassigned percept over the known and potential unions.The MAP inference on the other hand, just outputs the most probable union configuration.

In order for the binding inference to function properly, we have to define some hard rules (formulae with infinite weight) that reflect the binding restrictions in (2.8):

1. $\forall p \exists u : uniPer(u, p)$
2. $uniPer(u_1, p) \wedge uniPer(u_2, p) \Rightarrow u_1 = u_2$
3. $perFeat(p_1, f_1) \wedge perFeat(p_2, f_2) \wedge modPart(m, f_1)$
   $\wedge\, modPart(m, f_2) \wedge uniPer(u, p_1) \Rightarrow \neg uniPer(u, p_2)$

In the case of intra-modal subdivision of features with the predicate *typePart*, where percepts are restricted to just one feature per type (see Section 2.2.1), we can facilitate binding with an additional hard rule:

$$perFeat(p, f_1) \wedge typePart(t, f_1) \wedge typePart(t, f_2) \wedge (f_1 \neq f_2)$$
$$\Rightarrow \neg perFeat(p, f_2)$$

We can easily restrict this rule to specific feature types by grounding the variable t.

2.3 Binding as Part of a Cognitive System

One of the main challenges of cognitive architectures [42] is how to bridge the *semantic gap* between the sensory information and higher-level cognition. This problem is in a way very related to the *symbol grounding problem* [15]. The sensory data is first processed by a lower cognitive layer known as the *perceptual layer*. While processing on the perceptual layer is inherently *intra-modal*, and there is often little or no communication between the individual subsystems, the higher-level cognition usually assumes *a-modal* information. In this sense, binding plays an important role in overcoming the semantic gap, assuring that the resulting higher-level representations are *multi-modally grounded*.

If we model representations produced by the perceptual layer as *percepts* in the problem definition (Section 2.1), it is convenient to use *percept unions* as the basis for the higher-level representations. A-modal higher-level representations are thus grounded through a collection of modal percepts all the way to the sensory data.

Figure 2.2 illustrates a possible application of the binding system described in Section 2.2 to a cognitive system. We can see that three distinct processes use the information from the perceptual layer:

- The process of *concept grounding* uses modal concepts produced by the learning processes in modal learners (e.g. various colour and shape types) to ground the binding rules.

- The process of *instance grounding* relies on the ability of the perceptual layer to quickly present (usually relying on one modality only) quantitative estimates about the entities (instances) the cognitive system is currently sensing. While the multi-modal representations of perceived entities are quantitatively and qualitatively finalized by the binding process itself (union configuration), these initial approximative representations can be considered as a kind of placeholders for potential entities (i.e. possible percept unions). They are devoid of any features or other kind of attributes.

- The recognition process in modal learners results in the percept configuration, which represents the input to the process of *binding inference*.

The final product of binding – the union configuration is used both as the

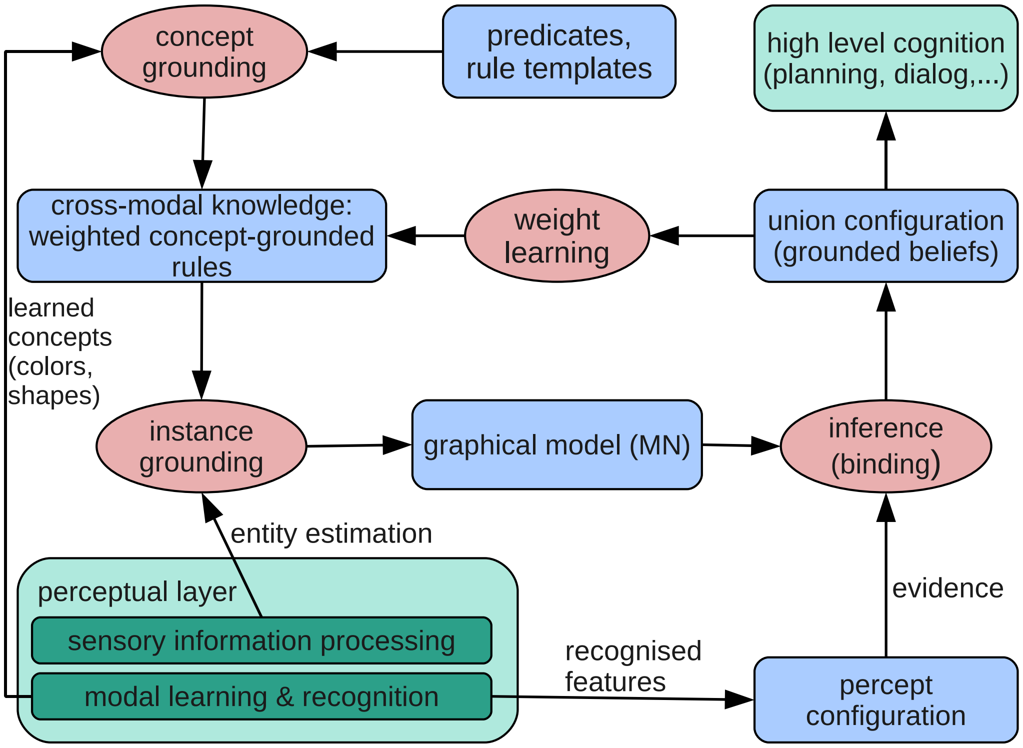
concept grounding
predicates, rule templates
high level cognition (planning, dialog,...)
cross-modal knowledge: weighted concept-grounded rules
weight learning
union configuration (grounded beliefs)
learned concepts (colors, shapes)
instance grounding
graphical model (MN)
inference (binding)
entity estimation
evidence
perceptual layer
sensory information processing
modal learning & recognition
recognised features
percept configuration

Figure 2.2: Cross-modal learning and binding as part of a cognitive system.

basis for a-modal representations in higher-level cognition, and as a source of learning samples for weight learning.

The processes of *instance grounding*, *binding inference* and *weight learning* form the *inner binding loop*, which exploits the perceptive abilities of modal learners and recognisers to improve its cross-modal associative power. On the other hand, the process of *concept grounding* exploits the concept forming ability of modal learners. By associating between modal concepts, it produces new cross-modal concepts, which the inner binding loop eventually evaluates within the existing cross-modal knowledge.

2.4 Real World Environment Issues

The real testing grounds for any cognitive architecture is of course a real world environment, i.e. real data in real time. Even if we neglect the qualitative aspect of real data (e.g. by assuming that it is completely handled by the perceptual layer), we simply can not ignore its quantitative aspect, i.e. the sheer numbers of features, entities and percepts that have to be processed by the binding mechanism and higher cognition in a limited time. Hence, scalability is an important requirement for every cognitive system operating in the real world.

The quantitative aspect of the real data in an MLN reflects in the size of the domain (constants) that is used to obtain the MN. Unfortunately, the size of an MN (number of grounded predicates and rules) can increase exponentially with the size of the domain – and size increases the inference time. Several generic solutions have been proposed to tackle the scalability problem of MLN, improving either the grounding process [28] or the inference itself [31].

A good implementation practice is to filter the formulae by the value of their weights before grounding them (the more the absolute weight of a formula approaches zero, the less relevant the formula is). Applied to our particular case this would result in an additional preprocessing step, prior to instance grounding that would purge the cross-modal knowledge base of irrelevant or immature associations. This would diminish the number of edges in the graphical model, thus relieving the binding inference of some processing burden. The weight learning would still affect the whole knowledge base. Furthermore, the cognitive system might even exploit the filtering mechanism to its advantage. Depending on situation, it could achieve faster response, or greater reliability and accuracy by regulating the filtering criterion.

In Section 2.1 we neglected the real-time issues, implying a somewhat static concept of percept configuration, where the system synchronously acquires percepts from different modalities. In the real-world environment, a percept configuration is rather subject to continuous smaller scale changes, typically involving only a small set of percepts from a single modality at a time. Substantial time differences in percept output between modalities can be expected even in the case of concurrent sensory stimuli. This scales down the original binding problem from Section 2.1. Rather than re-binding the whole percept configuration, the system only needs to establish how a fresh subset of percepts from a single modality relates to the existing union configuration. The real-world challenges could also prompt us to explore various possibilities of breaking the single graphical model implied in Section 2.1 into several smaller ones. For example, the system could first separately determine how a new percept relates to each of the existing percept unions and then, if necessary, combine only the most promising combinations to a single graphical model.

Chapter 3

George – a Prototype of a Cognitive System

In this chapter we present *George*, a robot prototype that implements a few of the most typical characteristics of cognitive systems. More comprehensive descriptions of George are available in [39, 37]. In brief, George is capable of active exploration (by turning its head) of its immediate surroundings, and of perception of small objects in them. It is also capable of making conversation about its perceptions with a human tutor, and of learning about objects' properties from that conversation.

George is the result of a joint research effort of six partners[1], within the FP7 European project *CogX*. Its purpose might be best described as a demonstration and evaluation platform for various cognitive paradigms, like visual attention, situated dialogue with another agent, learning through situated dialogue, motivation, planning, etc. (each of the partners involved in its creation, had its own particular agenda about what kind of cognitive mechanism to integrate in George). Thus, George represents the ultimate testing ground for such paradigms – a logical next step in evaluation after

[1] University of Birmingham (UK), DFKI (Germany), KTH Stockholm (Sweden), University of Ljubljana (Slovenia), Alfred Ludwig University of Freiburg (Germany) and TU Wien (Austria).

the classic off-line experiments (like the ones in Section 5.1).

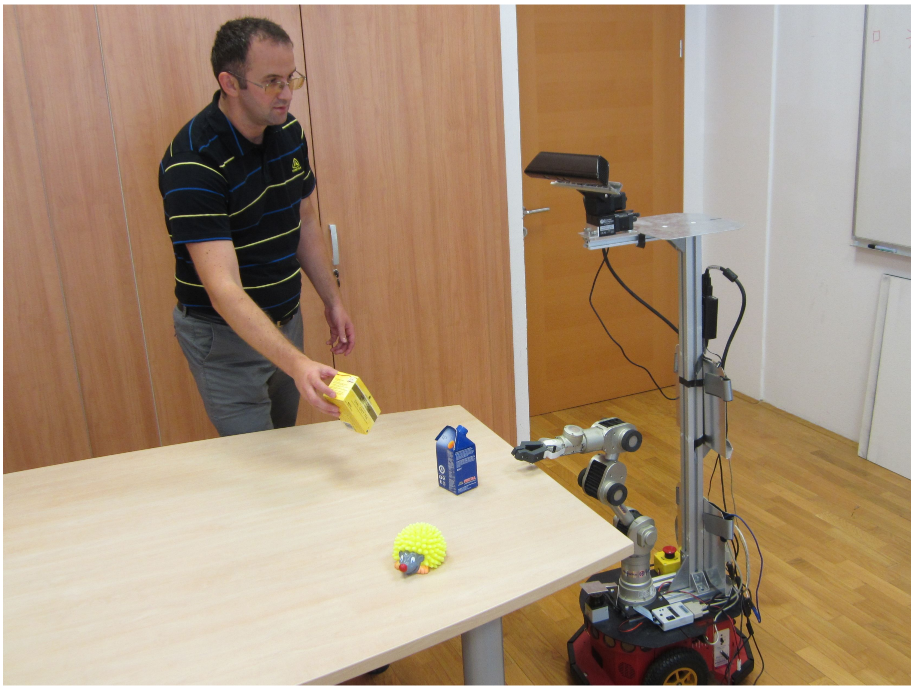

Figure 3.1: George in a learning interaction with a tutor.

During the years George (and his predecessor Playmate [45]) was built, an important lesson about artificial cognitive systems emerged: the key to a good cognitive system is the integration. Of course, it is not unexpected that we need some kind of solid integration to make various cognitive components work together. However, if we want to fully exploit the potential of each mechanism, and at the same time make them work as coherent unified system, in such case the complexity and the sophistication of the integration will probably have to exceed the cumulative complexity of individual components. The integration we are talking about, must not be understood in a technical sense, only, but also as the integration of paradigms, or sometimes even very basic ideas. Often, in the process of building George, it was necessary to modify assumptions or sacrifice important ideas about your cognitive paradigm in order to make the integration plausible. In this view, it is no surprise that we had to considerably adapt our binding platform from

Chapter 2, though the fundamental principles remain the same.

Thus, the purpose of this chapter is also to prepare the reader for what follows in Chapter 4, where we describe, how we adapted, upgraded and finally integrated into George the principles described in Chapter 2. The content of this chapter is a digest of a more detailed description of George available in [39]. We begin by describing the main competencies and representations that characterize the cognitive system. Section 3.2 focuses on the implementation and the integration of the system. Finally, in Section 3.3 we give an overview of the basic behaviour of the robot.

3.1 System Competencies and Representations

A robotic system for interactive learning in dialogue with a human must have the competencies to generate the required behaviour, including the ability to process representations stemming from different modalities. Figure 3.2 provides an overview of the main competencies in the George prototype system and the relationships between them. By processing visual information and communicating with humans, the system forms beliefs about the world. They are exploited by behaviour generation mechanisms that select actions to perform in order to extend the system's knowledge about visual concepts. In this section, we describe the individual competencies and representations required for interactive learning. To make these descriptions more concrete we first present an illustrative example, which briefly demonstrates the capabilities of the system, allowing us to ground later explanations in a real-world example.

3.1.1 An Example of Interaction

Consider a scene similar to the one presented in Figure 3.1. A human tutor and the robot system are engaged in a dialogue aiming to teach the robot

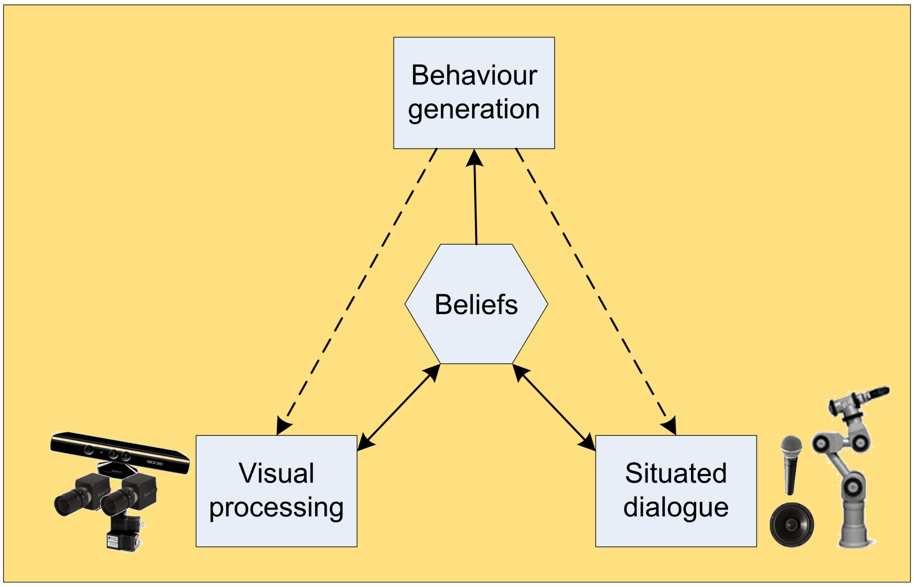

Figure 3.2: System competencies and relationships between them. Schema taken from [39].

about visual concepts, such as colour (e.g. red, blue, etc.), shape (elongated, compact) and object types (e.g. a mug, a bottle, etc.). The tutor puts the objects in the scene, and describes them or asks questions about them. In this scenario, all the perceptual entities (objects) are restricted to be a single shape, (predominant) colour and type. The robot looks around the scene, detects the objects, and processes the visual and linguistic information, thus trying to understand his environment. Based on his understanding of the scene, he plans how to learn more about the objects and their properties.

Let us suppose that the current view of the robot is depicted in Figure 3.3. The tutor may convey new information to the robot by describing one of the objects (e.g., H: 'The blue object is a bottle. It is elongated.'). After establishing common ground, by determining which object the human is referring to, the robot can update its knowledge about the concepts of "a bottle" and "elongatedness". The human may also ask the robot a question (e.g., H: 'Which colour is the tea box?'). The robot will answer the question (R: 'It is red.'). However, it could also take the initiative, and ask the tutor a question that would require an answer. That would increase its knowledge about the objects currently perceived in the scene, and about object properties in general (R: 'What shape is the cornflakes box?', or R: 'Please, show me something green.'). The robot can also point at an object to avoid ambiguous questions; e.g., since there are two mugs, and two red and two yellow objects in the scene, the robot can not refer to one of the mugs verbally, so it would point at one of them to establish the common ground. Only then it would ask a question like R: 'What shape is this object?'. In such a *mixed initiative dialogue*, the robot tries to get as much information from his tutor as possible to learn about objects and their properties. In the remainder of this section, we will describe the competencies and representations that facilitate these kinds of behaviour.

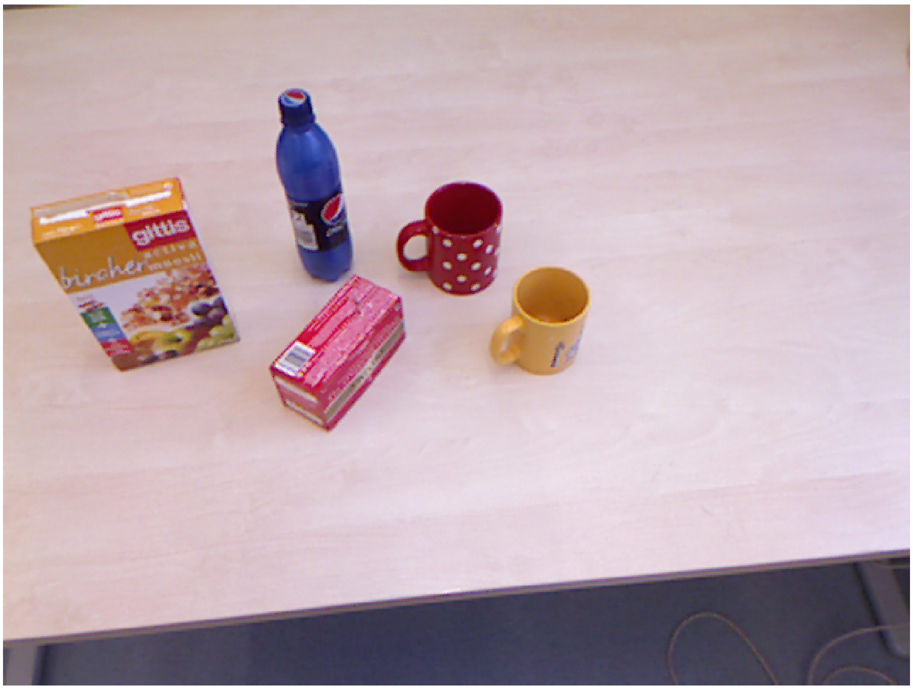

Figure 3.3: An example of a scene George can learn from.

3.1.2 Attention Driven Vision[2]

To learn autonomously about visual object concepts, the robot has to be able to detect new objects, when they are presented, as well as identify interesting parts of the scene. Since the robot can not have models for unknown objects, it can not rely on model based detection and recognition. Instead, it requires a more general mechanism. Hence, George relies on a generic, bottom-up 3D attention mechanism for object detection.

To make the problem of generic segmentation of unknown objects tractable, the system assumes that objects are always presented on a supporting surface such as the table in Figure 3.3. Given 3D point clouds that are obtained with an RGB-D sensor, the system detects supporting planes using a variant of particle swarm optimization [49, 50]. Any point clouds sticking out from the detected supporting planes are labelled as 3D *spaces of interest* (SOIs), i.e. something that is potentially interesting to the system (in the case of Figure 3.3, the robot would detect five different SOIs). Using their position, size and colour histogram, the system can track over time individual SOIs, thus eliminating transient features or noise.

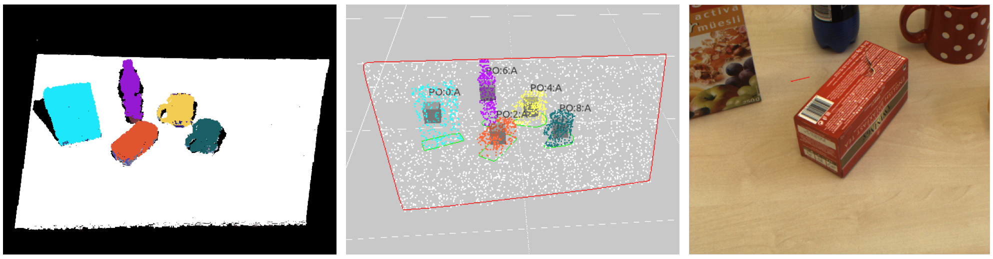

Figure 3.4: Segmented point cloud, detected objects, and a close-up view of a foveated object.

A segmentation that is based on the RGB-D data, only, is not reliable. In our case, it may include points with erroneously assigned background colour, due to shadowing effects at object boundaries. Therefore, by using the *graph cut* ([4]) method, SOIs are supplemented with a precise segmentation mask.

[2] This section abridges Section 2.2 in [39].

This segmentation happens in a foveated (i.e. higher-resolution) view of the potential object, using an RGB image from a camera with a longer focal length than the RGB-D sensor. The object features, used for learning visual properties, are extracted based on this segmentation mask (e.g. the medians of the HSL colour values of all foreground pixels, different shape features, etc.). Figure 3.4 shows the results of processing the scene depicted in Figure 3.3: the segmented point cloud, the detected objects and the close-up view after foveating on an object. Segmented objects are then subject to individual processing.

3.1.3 Learning and Recognizing Object Properties[3]

To efficiently store and generalise visual information, the visual features of object properties (such as colours and basic shapes) are internally represented as generative models. These generative models take the form of probability density functions (PDFs) over the feature space, and are constructed in an on-line fashion from new observations. This continuous learning process extracts visual data in the form of multidimensional features from the segmented objects (e.g. features relating to shape, texture, colour and intensity). The *on-line discriminative Kernel Density Estimator* (odKDE [25]) gradually constructs estimations of the PDFs in this feature space. A particularly important property of the odKDE is that it allows adaptation of the models from both, the positive and the negative examples (i.e. learning and unlearning; [26]).

During its on-line operation, the system maintains a multivariate generative model (e.g. over HSL colour feature space) for each one of the visual concepts (e.g. every colour) that were already introduced. Furthermore, for mutually exclusive sets of concepts (e.g. all colours), the optimal feature subspace is continually determined by feature selection. This feature subspace is used to construct a Bayesian classifier for individual object properties.

In addition, the system maintains an “unknown model”, which accounts

[3] This section is a digest of Sections 2.3 and 2.4 in [39].

for poor classification when none of the current concept models supports the last observation strongly enough. Having built such a knowledge model and Bayesian classifier, recognition is done by inspecting the a-posteriori probabilities over individual concepts and the unknown model.

By analysing the a-posteriori probability, the system is able to determine the *information gain* for every concept. The information gain for a concept estimates how much the system would increase its knowledge if it were to receive information from the tutor about that concept with respect to a given object. This serves as a basis for triggering *situated extrospective* learning mechanisms.

Furthermore, even in the absence of visible objects, the system can inspect its models and determine which model is the weakest or most ambiguous. Based on this estimate, the information gain for every concept is again calculated, regardless of what is visible. This measure is used to initiate *introspective* learning.

Besides generic object properties, George is also able to recognise and learn object types. The method [51] combines appearance based (RGB image) and shape based (point cloud) visual features into multi-view object models. The views are incrementally acquired from RGB-D images, and are aligned using sparse bundle adjustment. Type recognition uses RANSAC to find a matching view from the features extracted from the object RGB image and point cloud. Even though what is essentially an object instance based recogniser is used as a type classifier, this works well enough in handling the variability encountered in George scenarios.

3.1.4 Situated Dialogue[4]

The other external source of information for George is its tutor. In task-oriented interactions between a human and a robot, there is more to dialogue than just understanding words. The robot not only needs to understand *what* is being talked about, but also *why* something was told. In other words, what

[4] This section is a digest of Section 2.5 in [39].

the tutor *intends* the robot to do with the information in the larger context of their joint activity (e.g. which part of information received from the tutor is intended for learning).

An *intention* is a goal-oriented cognitive state, usually modelled as an explicit commitment to acting in order to achieve a goal or desire [5, 8]. George's communication system explicitly models *communicative intentions* (i.e. intentions that are related to communication, as opposed to the robot's purely *internal* intentions or goals; see Section 3.1.6), and uses them as a pragmatic representation of the human-robot interaction, abstracting away from the actual surface form.

The system employs *continual abduction* ([22]) to generate and verify hypotheses about the tutor's behaviour in terms of communicative intentions. Abduction is a method of explanatory logical reasoning ([11]). Given a theory T, a rule $T \vdash A \rightarrow B$ and a fact B, abduction allows inferring A as an explanation of B. B can be deductively inferred from $A \cup T$. If $T \nvdash A$, then we say that A is an *assumption*. There may be many possible causes of B besides A. Abduction amounts to *guessing*; assuming that the premise is true, the conclusion holds too.

Abductive reasoning over intentions in a situated context is a bi-directional process ([41]) that the system uses in two roles: *recognition* of the *tutor's* communicative intentions (inferring their intention given the context and a surface form of their input); and *realisation* of the *robot's* communicative intentions (inferring an appropriate surface form given the context and the robot's intention).

3.1.5 Higher-Level Representation with Beliefs

By processing visual information and communicating with the human, the system forms *beliefs* about the world. Beliefs are data structures that contain indexical information about perceived entities. They form a representational layer where multi-modal and multi-agent information is associated and merged to create a-modal representations – a process akin binding. In general, we can regard a belief as a higher-level representation of an element of the physical reality, potentially grounded in one or more sensory inputs, and attributed to specific agents or groups. In George, a single belief contains information about a single entity, but there can be many beliefs about the same entity. The contents of beliefs are expressed as multivariate probability distributions over feature-value pairs. We provide a more detailed account of the belief layer in Chapter 4.

3.1.6 Motivation[5]

In order to discover and make sense of its surroundings, George has to perform multiple, possibly interleaved, goal-directed activities. As a cognitive system that must fill gaps in its own knowledge, it is important that it is able to generate and manage its own goals, since the opportunities available to it at runtime may be unknown or unpredictable at design-time. The *motivation framework* [16, 47] encodes the *drives* of the system (the general types of things it wants to achieve) as a collection of *goal generators*. Each of them generates particular types of goals for the system based on the current belief state and communication with the tutor. A goal is a description of a desired future situation (e.g. to know the colour of a newly visible object).

The goal generators in George create goals necessary to engage in situated dialogue with a human tutor and to learn about its surroundings. A particular goal generator monitors the communicative intentions provided by the dialogue subsystem as it interprets tutor's utterances (see 3.1.4). Based on their content, this generator creates goals to answer questions about objects, or to perform learning. Each of these goals contains a reference to the merged belief representing the referred object, and other intention-specific information. Another goal generator handles the situation where a set of possible intentions has been generated in response to an ambiguous reference. In this case, the goal not only includes the content describing the future state, but

[5] This section abridges the part of Section 2.8 in [39] that describes the motivation system.

an existentially qualified reference to a belief that represents the possible referents of the intention. Part of George's task is then to resolve this reference before it can act on its content.

Each potential goal has to pass through a management system that determines which of them should be pursued, i.e. activated. The aim of this filtering step is to prioritise important or more appropriate goals in given situation. The system activates goals based on a *priority hierarchy of drives.* Each of three drives in this hierarchy controls a type of behaviour that George was designed to perform. The highest priority drive is to *answer tutor's questions*, followed by the drive to *learn by extrospection* (i.e. inspecting the world external to the agent). At the lowest level, we find the drive to *learn by introspection.* Goals of a particular priority suppress the activation of goals with lower priorities. Goals that pass through this filter are ranked according to heuristics provided by their goal generators. The top ranked goals are activated.

3.1.7 Planning[6]

Once goals are activated, it is up to the planning system, how to achieve them in the current context. An active goal and the current context (which is derived from the belief state) form a planning problem description. Plan execution, execution monitoring and replanning is managed via a collection of action interfaces that trigger individual components in the modality-specific subsystems. The planning system is based on the *fast downward planner* [19]. This is a state of the art planning system based on heuristic forward search, extended by a preprocessing routine, which enables the support of object fluents and numerical constants by compiling them away, and deal with the uncertainty of the real-world environment by using a continual planning approach ([6]).

The dialogue with a tutor plays a central role in the George scenario. Asking and answering questions is crucial for all three George's drives. Hence,

[6] This section abridges the part of Section 2.8 in [39] that describes the planning system.

the planner must generate plans that establish a common ground with the tutor about the object of their discussion. For instance, a possible initial ambiguity in a dialogue is represented by having multiple objects as the referents of a tutor's question, alongside a goal to only have a single referent. The planner can predict the effects of available *clarification actions* on its interpretation of that reference. It uses these actions to create a plan, which it expects to remove the ambiguity and leave only a single referent that will be the next topic of the conversation.

George can choose between two types of actions for clarifying a reference: describing the object verbally, or pointing at it with its arm. Since a verbal description is considered cheaper, George will always try to describe the object first, if it has a combination of recognised properties that is unique among the perceived objects, and can be verbalised. Otherwise, it will resort to pointing. Once a common ground is established, the planner will determine a suitable answer or question from the belief state, and trigger learning, if necessary. Examples of behaviour generated by George's planning system are available in Section 5.2.

3.2 Implementation and Integration

To make the competencies described above work together in a robotic system, a sophisticated means of integration is needed. The implementation and integration of George is based on CAST, the CoSy Architecture Schema Toolkit [17]. CAST is a distributed working-memory model composed of several sub-architectures, each implementing a different functionality. A sub-architecture (SA) contains one or more components each running in its own thread. The components communicate through a working memory (WM). When the state of a component changes, it either adds an entry (of a certain type) containing the relevant information to the WM, updates an existing entry in the WM, or deletes it from the WM. Another component can register with the WM to receive notifications whenever changes to entries of a certain type occur. This allows links between multiple components to be established, and for information to be passed accordingly. The architectural approach is described in more detail in [17, 18].

George is composed of six CAST SAs. Figure 3.5 presents George from the system point of view. It illustrates its complexity, and denotes the relations between individual components.

The *Visual SA* is responsible for visual processing. It implements competencies described in Sections 3.1.2 and 3.1.3. The SA uses a Kinect RGB-D

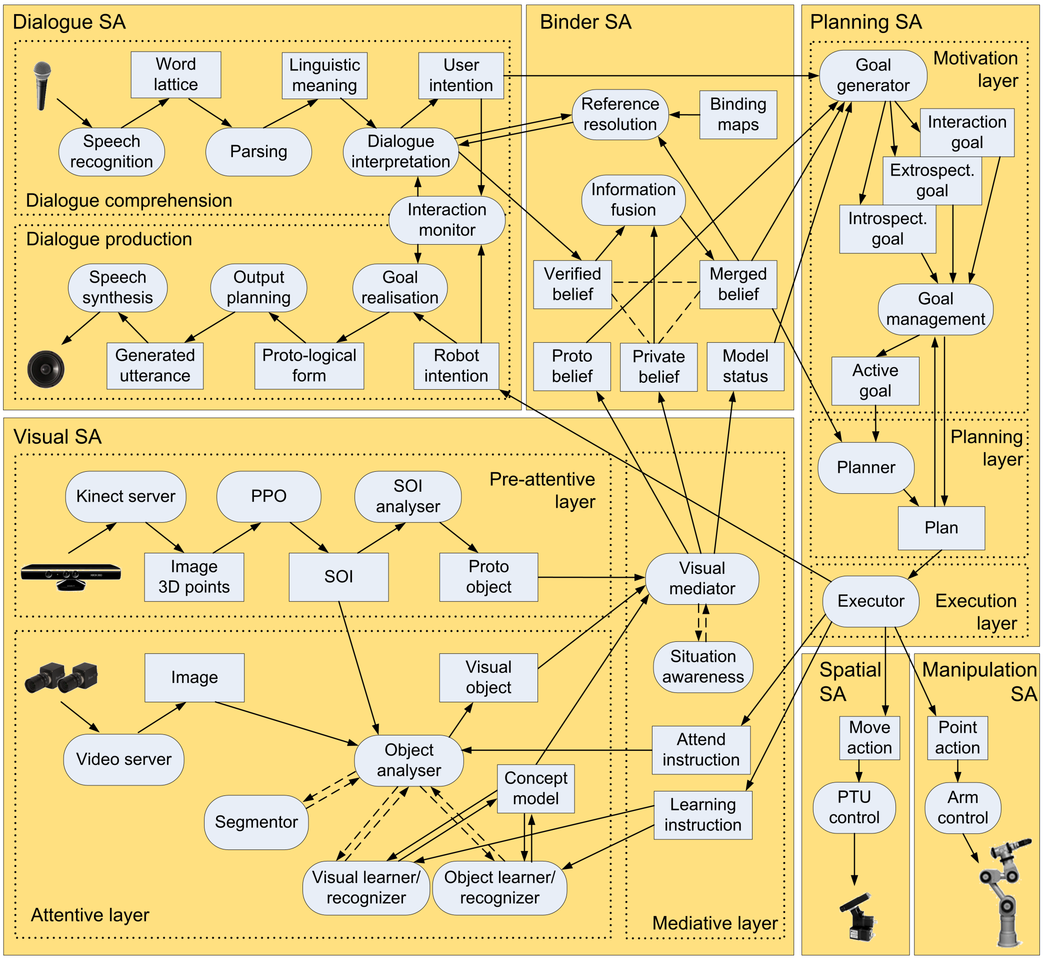

Figure 3.5: A schematic view of George system architecture. Rounded boxes represent the components, rectangles represent the data structures, while arrows indicate information flow. Schema taken from [39].

sensor for pre-attentive vision, i.e. identifying and processing spaces of interest (SOIs) as they appear in the scene. On the other hand, it uses a narrow field-of-view Point Grey Flea 2 camera for attentive vision, i.e. deliberate extraction of object properties. The attention mechanism also makes use of the Direct Perception pan/tilt unit (part of *Spatial SA*) for bringing SOIs into the centre of view.

The *Dialogue SA* provides the functionality for the situated dialogue. The system uses a third party software for speech recognition, and the Mary TTS system for speech production[7]. The SA implements techniques presented in Section 3.1.4 for recognition of the tutor's intentions and realisation of robot's intentions in the situated context. The robot also uses the Neuronics Katana 6M (5 DOF) robotic arm for pointing at objects. The arm is controlled via Golem [24], and is part of *Manipulation SA*).

Beliefs are collected in the *Binder SA*, which represents a central hub for gathering information about perceptions from different modalities (sub-architectures). The *Planning SA* monitors the beliefs, and generates appropriate behaviour as described in Section 3.1.6. As new beliefs appear, they trigger goal generators to produce planning goals, while the overall set of current beliefs represent the planning state. During a plan execution, requests are sent to the Visual, Spatial, Manipulation and Dialogue SAs to perform planned actions, which generates the desired behaviour.

3.3 Basic Behaviours

George is a very complex and heterogeneous cognitive system. This means that even its basic mechanisms of behaviour combine functionalities distributed across several sub-architectures, hence the need for a tight and meaningful integration of its components. Very often, those behaviours require that different functionalities are executed in parallel, although in a synchronised manner. The cognitive mechanisms that implement George's

[7] http://mary.dfki.de

behaviour can be grouped in four main groups:

- mechanisms for visual perception,
- tutor initiated interaction,
- extrospective learning mechanisms,
- introspective learning mechanisms.

Two mechanisms provide the robot with the visual information. The first one is bottom-up, and is triggered by changes in the scene. It makes sure that the robot analyses the objects that are brought in his view. The second one is top-down, and is triggered by the motivation subsystem. It makes the robot explore the scene, using its pan-tilt unit, searching for new objects that were possibly introduced to his surroundings, while it was looking away.

Interaction with a human tutor is one of the crucial abilities of this cognitive system. Interaction can be triggered by the tutor or by the robot. The tutor can trigger interaction in three ways: by asking the robot to execute an instruction, by asking robot a question (e.g., 'What colour is the coke can?' or 'Is the coke can red?'), or by giving the robot useful information (e.g., 'The coke can is red.') that can be used for learning (*situated tutor-driven learning*). These mechanisms are triggered by the system's *interaction goals.*

Of course, a cognitive system can not just passively wait for tutor's learning instructions, it has to exploit learning opportunities. It should actively look for, ask for, and use the information that would help to extend its knowledge. These learning mechanisms associated with such behaviour (*the situated autonomous learning* and *the situated tutor-assisted learning*) are part of George's *extrospective drive.* The goal generators monitor the beliefs for information that can be exploited for learning. They generate a goal for each possible learning opportunity.

In the absence of opportunities for situated learning, the robot can still actively engineer interactions to provide new information. E.g., the robot can autonomously search for new objects, or even ask another agent to provide one (specifying the properties that are the most interesting). This behaviour is based exclusively on introspection of the existing property models. From a pool of currently maintained models, the robot selects the one it considers the least adequate (typically inadequately sampled), and initiates an action that tries to obtain new samples to improve it.

Chapter 4

Bridging the Semantic Gap in George

As we saw in Chapter 3, the two critical properties of a cognitive system operating in a complex environment are (i) the ability to sense, perceive and process complex information about physical reality, and (ii) the ability to use this information to plan, manage and execute complex actions in such an environment. The complexity of the physical reality implies the ability to collect information from different sources, i.e. different sensor types and possibly different agents (other than the cognitive system itself). This means that at least on lower levels the information is inherently multi-modal and multi-agent. On the other hand, the higher cognition (e.g. motivation, planning, etc.) predominantly assumes a-modal information. Hence, an intermediate cognitive layer capable to relate and merge multi-modal and multi-agent information is needed to close the *semantic gap* that divides the lower and the higher cognition.

In this chapter, we describe in more detail the intermediate cognitive layer of the prototype cognitive system George. As we saw earlier, the scenario assumes a robot (George) capable of making situated dialogue with a human tutor about objects on a table. The robot is thus able to observe, track and recognise the objects on the table, and through the dialogue with the tutor improve its knowledge about the objects' properties (cross-modal learning). Such a scenario obviously relies on the ability of the robot to first associate, and later merge multi-agent information. Higher-level cognitive processes, like motivation and planning, can then use the resulting representations.

4.1 Reference Resolution

The process of determining the denotation of a referring expression is called *reference resolution.* In a cognitive system, *reference resolution* can be formulated as a process akin to binding that tries to associate multi-agent information. Both processes can operate on the same cognitive layer, called *the belief layer*, briefly introduced in Section 4.2. As we will see later, in Section 4.3, the difference between robot's own perceptions and information attributed to another agent both encoded in beliefs can be directly exploited for implicit learning.

In general, the binding in Markov Logic Networks is applied to an intermediate cognitive layer, where the various beliefs represent perceived and assumed facts. These beliefs are used to instantiate the rules from the cross-modal knowledge base to a Markov graphical model. We saw in Section 2.3 that MLN knowledge represents the general rules encoding relations between concepts (e.g. object properties as colour, shape,etc.), while a graphical model encodes the relations between concrete instances (objects) that are currently perceived by the system. A successful inference results in a shared multimodal representation of a physical entity. Such representations can be used as learning opportunities to improve cross-modal knowledge.

In George, the principles of binding are used for reference resolution. We base our implementation of this process on the general method of binding in Markov logic networks described in Section 2.2. In our case, the robot uses reference resolution to relate its own perceptions to information attributed to a human tutor. Hence, reference resolution is critical for its ability to make situated dialogue with the human.

We implemented the MLN as a set of components that process information stored in beliefs (see Figure 4.1). A MLN engine component maintains a Markov network graphical model, which makes continuous on-line inference (MCMC sampling), and can continuously adapt to the changes in beliefs. MLN engines can also combine the information encoded in the current graphical model with external information about the correct inference outcome to perform on-line weight learning. MLN client components filter the information stored in beliefs or other data structures, and feed it to MLN engines. They can also read and process inference results, and trigger weight learning in MLN engines.

The implementation of reference resolution in George features a single MLN engine and two MLN clients. One MLN client (the belief filter) continuously filters the information in beliefs, and forwards it to the MLN engine as evidence about perceived entities. The other MLN client (the restrictor) acts on request; triggered by the dialogue subsystem (when it recognises a referring expression in the tutor's utterance) it first feeds the MLN engine with the referring (restrictive) information, then reads and forwards the result of the inference back to the dialogue subsystem, and finally withdraws the referring information.

The result of the MLN inference is a probability distribution over perceived entities, represented as beliefs. The dialogue subsystem uses it to determine the interpretation of the tutor's utterance. This eventually results in additional beliefs related to the ones grounded in robot perceptions (see Section 4.2). As a consequence of successful reference resolution, the restrictor can also trigger weight learning in the MLN engine. A successful reference resolution usually means that the resulting probability distribution favours with a suitable degree of reliability the denotation to one of the existing beliefs. In this case, the restrictor first feeds the 'winning' resolution to the MLN engine as evidence, and then triggers the learning. Afterwards, it withdraws both pieces of evidence, the referring information and the 'winning' resolution, from the MLN engine.

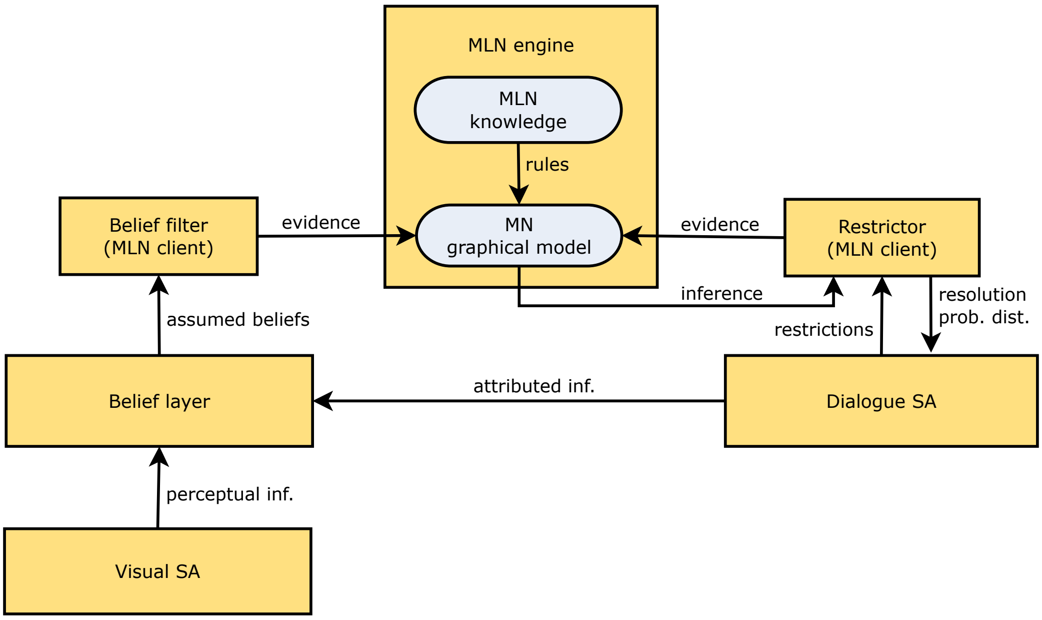

Figure 4.1: Implementation and integration of reference resolution in George.

4.1.1 An Example of Reference Resolution

The following is an example of reference resolution performed in an MLN engine component. We assume a small MLN reference resolution knowledge database that encodes associations between two visual colour models (denoted as $Color1$ and $Color2$) and two linguistic colour descriptions (Red and $Blue$):

$$\begin{array}{rl}
2.5 & percColor(b, Color1) \wedge restrict(Red) \Rightarrow resolveTo(b) \\
-1.9 & percColor(b, Color1) \wedge restrict(Blue) \Rightarrow resolveTo(b) \\
-1.3 & percColor(b, Color2) \wedge restrict(Red) \Rightarrow resolveTo(b) \\
2.0 & percColor(b, Color2) \wedge restrict(Blue) \Rightarrow resolveTo(b)
\end{array}$$

The predicate $percColor(b, Color1)$ denotes that the object represented by the belief b was perceived to be of modal colour representation $Color1$ by the visual subsystem. The predicate $restrict(Red)$ denotes Red as the restriction (referring information, see Section 4.2.1) given by the tutor, while the predicate $resolveTo(b)$ denotes the reference resolution to the belief b. As is the case in Section 2.2, variables begin with a lowercase character, while constants begin with an uppercase character.

We can see that the rules in the knowledge database instantiate concept (in our case the colours), but encode beliefs about objects as variables. The predicate $resolveTo(b)$ is also the object of the MLN engine query. The inference can result in the following probability distribution, for example:

$$\begin{array}{ll} 0.2 & resolveTo(B1) \\ 0.1 & resolveTo(B2) \\ 0.7 & resolveTo(B3) \end{array}$$

As was the case in Section 2.2.1, the real numbers denote the probabilities. In addition to the knowledge database, the reference resolution system includes the following set of hard rules (similar to the binding restrictions in Section 2.2.3) that regulate the inference process:

1. $belief(b_1) \land belief(b_2) \land resolveTo(b_1) \land resolveTo(b_2) \Rightarrow b_1 = b_2$
2. $resolveTo(b) \Rightarrow belief(b)$
3. $resolveTo(b) \Rightarrow \exists f : restrict(f)$

The hard rules are rules with an infinite weight that can never be broken. The predicate $belief(b)$ denotes the existence of belief b. Rule 1 restricts the reference resolution to exactly one belief, rule 2 restricts the reference resolution to an existing belief, and finally, rule 3 makes reference resolution possible only when referring information exists.

Let us suppose the system perceives two objects on the desktop, one red (perceived as $Color1$) and one blue (perceived as $Color2$). The belief filter feeds the MLN engine with the following evidence:

$$belief(B1) \land belief(B2) \land percColor(B1, Color1) \land percColor(B2, Color2)$$

Based on this information the MLN engine builds a Markov Network graphical model (MN). First it instantiates the rules with both beliefs:

$$\begin{aligned}
2.5 &\quad percColor(B1, Color1) \wedge restrict(Red) \Rightarrow resolveTo(B1)\\
-1.9 &\quad percColor(B1, Color1) \wedge restrict(Blue) \Rightarrow resolveTo(B1)\\
-1.3 &\quad percColor(B1, Color2) \wedge restrict(Red) \Rightarrow resolveTo(B1)\\
2.0 &\quad percColor(B1, Color2) \wedge restrict(Blue) \Rightarrow resolveTo(B1)\\
2.5 &\quad percColor(B2, Color1) \wedge restrict(Red) \Rightarrow resolveTo(B2)\\
-1.9 &\quad percColor(B2, Color1) \wedge restrict(Blue) \Rightarrow resolveTo(B2)\\
-1.3 &\quad percColor(B2, Color2) \wedge restrict(Red) \Rightarrow resolveTo(B2)\\
2.0 &\quad percColor(B2, Color2) \wedge restrict(Blue) \Rightarrow resolveTo(B2)\\
\infty &\quad \neg resolveTo(B1) \vee \neg resolveTo(B2)\\
\infty &\quad \neg resolveTo(B1) \wedge \neg resolveTo(B2)
\end{aligned}$$

Then it applies the evidence to the instantiated rules:

$$\begin{aligned}
2.5 &\quad restrict(Red) \Rightarrow resolveTo(B1)\\
-1.9 &\quad restrict(Blue) \Rightarrow resolveTo(B1)\\
-1.3 &\quad percColor(B1, Color2) \wedge restrict(Red) \Rightarrow resolveTo(B1)\\
2.0 &\quad percColor(B1, Color2) \wedge restrict(Blue) \Rightarrow resolveTo(B1)\\
2.5 &\quad percColor(B2, Color1) \wedge restrict(Red) \Rightarrow resolveTo(B2)\\
-1.9 &\quad percColor(B2, Color1) \wedge restrict(Blue) \Rightarrow resolveTo(B2)\\
-1.3 &\quad restrict(Red) \Rightarrow resolveTo(B2)\\
2.0 &\quad restrict(Blue) \Rightarrow resolveTo(B2)\\
\infty &\quad \neg resolveTo(B1) \vee \neg resolveTo(B2)\\
\infty &\quad \neg resolveTo(B1) \wedge \neg resolveTo(B2)
\end{aligned} \tag{4.1}$$

The instantiated rules above represents a MN, where each fully instantiated predicate represents a sampling variable (atom). The MLN engine performs a continuous inference that in the present case (because of the last instantiated rule, derived from the hard rule 3) does not yield any positive resolution.

Now, let us suppose that a human refers to a red object in his utterance. By the request of the dialogue subsystem, the restrictor component feeds the MLN engine with the predicate *restrict*(*Red*). This new piece of information modifies the graphical model as follows:

$$\begin{array}{rl}
2.5 & resolveTo(B1) \\
-1.9 & restrict(Blue) \Rightarrow resolveTo(B1) \\
-1.3 & percColor(B1, Color2) \Rightarrow resolveTo(B1) \\
2.0 & percColor(B1, Color2) \wedge restrict(Blue) \Rightarrow resolveTo(B1) \\
2.5 & percColor(B2, Color1) \Rightarrow resolveTo(B2) \\
-1.9 & percColor(B2, Color1) \wedge restrict(Blue) \Rightarrow resolveTo(B2) \\
-1.3 & resolveTo(B2) \\
2.0 & restrict(Blue) \Rightarrow resolveTo(B2) \\
\infty & \neg resolveTo(B1) \vee \neg resolveTo(B2)
\end{array}$$

As we can see, the referent information also removes the hard rule preventing any positive reference resolution. The inference result is now clear; the resulting probability distribution reliably indicates the belief $B1$ as the referent. The restrictor forwards this information to the dialogue subsystem and removes the referent information, which returns the graphical model to the state in (4.1).

Saliency can be a very useful addition to the situated human-robot dialogue. An object on the desktop can become salient because of non-verbal communication (e.g., the robot or the human pointing with his arm, or directing his gaze towards an object), or simply by being the only object on the desktop. The information about the saliency has to be part of the belief representing the object. The belief filter can feed this information to the MLN engine simply as the predicate $salient(B1)$. The human can then refer to that object with the word ‘this’, which the restrictor can represent with the predicate $restrict(This)$. The easiest way to implement this mechanism is to add another rule to the regulative set of hard rules:

$$salient(b) \wedge restrict(This) \Rightarrow resolveTo(b).$$

When the human refers to a salient object with the word ‘this’, the (instantiated) hard rule above simply overrules all the instantiated soft rules in the graphical model, resolving the reference to the salient object.

4.2 The Belief Layer

Beliefs form a cognitive layer where multi-modal and multi-agent information is associated and merged to a-modal representations. In general, a belief can be regarded as a higher-level representation of an element of the physical reality, which is grounded in one or more sensory inputs, attributed to a specific agent, or a combination of both. Our belief scheme distinguishes five distinct belief categories:

- *Private beliefs* reflect the robot's perceptions of the environment based on its sensory input. Private beliefs are expressed in modal symbols and can form various associations with private beliefs stemming from other modalities or beliefs with other epistemic statuses.

- *Assumed beliefs* are used to establish cross-agent or cross-modal common ground. They are created from private beliefs by translating the modal symbols to the a-modal ones. Depending on complexity of the modal learners and their ability for autonomous unsupervised learning, this process can be as simple as one-to-one symbol mapping or much more complex (e.g. translating between two sets of symbols with overlapping meaning that consequently also modifies the original probability distribution). In cross-agent case, the robot uses assumed beliefs to establish a common ground with another agent to facilitate communication. Thus, the beliefs reflect the robot assumptions about the meaning of its perceived information for a particular agent (e.g. human). In cross-modal case, the assumed beliefs establish a common ground between modalities. In both cases, this process facilitates cross-belief information fusion in later stages.

- *Attributed beliefs* contain information that robot attributes to another agent (e.g. human). This kind of beliefs are the direct consequence of some kind of communication with another agent. The robot is in principle able to analyse and understand the information in such beliefs, but does not necessarily agree with it (especially, if it does not match its own perception of the same reality).

- *Verified beliefs* are created from *attributed beliefs*. They contain the acknowledged information from the attributed beliefs. Acknowledgement (or verification) does not necessarily mean that the agent's information in the belief is consistent with the robot's perception; it just means that that information was adequately processed, and is now ready to be used in higher-level cognition (e.g. in communication with the agent that issued it). After a successful reference resolution, the restrictive information is stored in verified shared beliefs, while the asserted information is in attributed belief.

- *Merged beliefs* combine information from *verified* and *assumed beliefs*, and represent the final a-modal situated knowledge, ready to be used by the higher level cognitive processes (like motivation and planning). They contain as reliable information as possible, and as much information as available. Information can be merged in different ways. For example, the system can completely trust a certain agent (typically a tutor), so that the merged belief contains all information from the verified belief, and only uses the assumed belief to fill the information gaps left by the verified belief. A more complex solution for the information fusion involves merging probability distributions over feature values. The merging process is also called *information fusion*.

Private beliefs are created by mediator components, using the information from modal subsystems. On the other hand, attributed and verified beliefs are products of successful resolutions of another agent's references. The changes in perception propagate in real-time through the belief structure, from private beliefs to the merged ones. In a similar manner, the progress in dialogue and dialogue processing (certain events in other subsystems can be treated as acknowledgements for the attributed information) are reflected in changes in attributed and verified beliefs. This means that the process of information fusion (belief merging) has to be repeated each time new perceptual information propagates to the assumed belief, or new attributed information is verified.

4.2.1 An Example of Information Flow in Beliefs

Figures 4.2, 4.3 and 4.4 illustrate how belief representations of an object change with the activity of the system. Objects are described in terms of colours, shapes and affordances[1]. The goal of the system is to use the new information provided by a human tutor for visual learning.

Figure 4.2 represents the state of beliefs after the robot has processed the visual information about a physical object on the desktop. It reflects the robot's own perception of the object. We can see that the internal (modal) visual symbols (and the object's affordance, which is based on its shape) are translated to a-modal symbols (in our case the dialogue subsystem also operates with a-modal symbols). The translation can be performed by an MLN engine component. As described in Section 4.1, the translation can be more than just a simple symbol mapping; it can also have to re-calculate the probability distributions of the translated symbols. The merging process in this case just forwards the information to the merged belief.

Figure 4.3 represents the structure of beliefs after the system has processed a tutor's statement about the object ("The compact object is blue."). In this sentence the 'compact' represents the referring or restrictive information, which is used to determine (restrict) the entity in question. With the assertive information in the sentence ('blue') the human expressed a new quality about the referred entity (perhaps not known to the robot). The assertive information does not completely agree with the robot's perceptions. Fortunately, the restrictive part of the statement is consistent with the analogue information in the current merged belief, which guarantees the success of the reference resolution. We can see that the information attributed to

[1] An affordance of an object is the possibility of an action to be performed on that object by an agent [13].

the tutor is initially split in two parts. The restrictive part is already considered verified (since the reference resolution was successful), and goes to the verified belief, while the assertive part goes to the attributed belief, since it is not yet clear, whether it represents a common ground between the robot's and tutor's perceptions (we can see that in our particular example the doubt is justified). We can see that the merging process confirms the shape information in the merged belief.

Figure 4.4 illustrates what happens after a certain event in other parts of the system (in our case the visual learning) triggers the acknowledgement (of a portion) of the asserted information. The acknowledged attributed in-

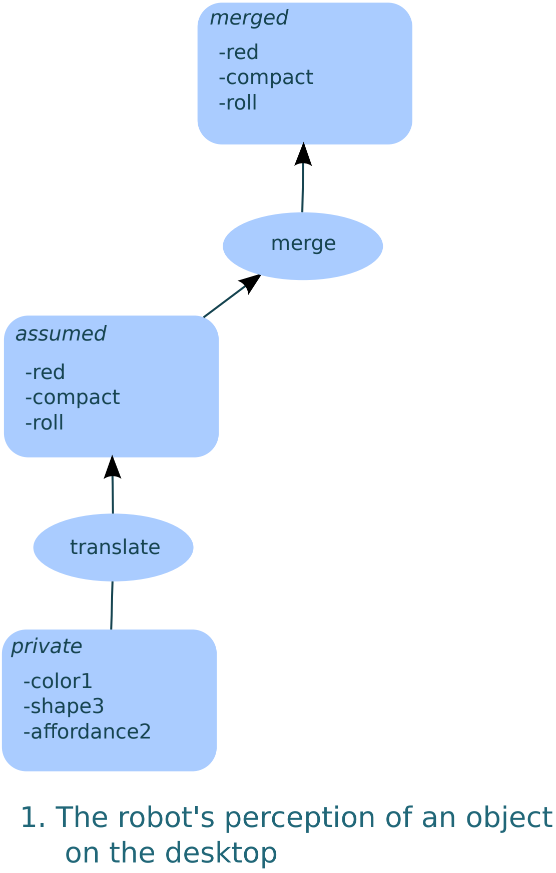

Figure 4.2: A structure of beliefs reflecting the robot's own perception of an object.

formation is propagated to the verified belief, and then merged. In our case, the colour property is replaced with its attributed version. The merged belief therefore contains one piece of information that is purely perceptual ('roll'); the information about the shape ('compact') is shared by both, robot's perception and human's description; the colour information ('blue') is not shared, but since it is provided by the tutor, it is treated as more reliable.

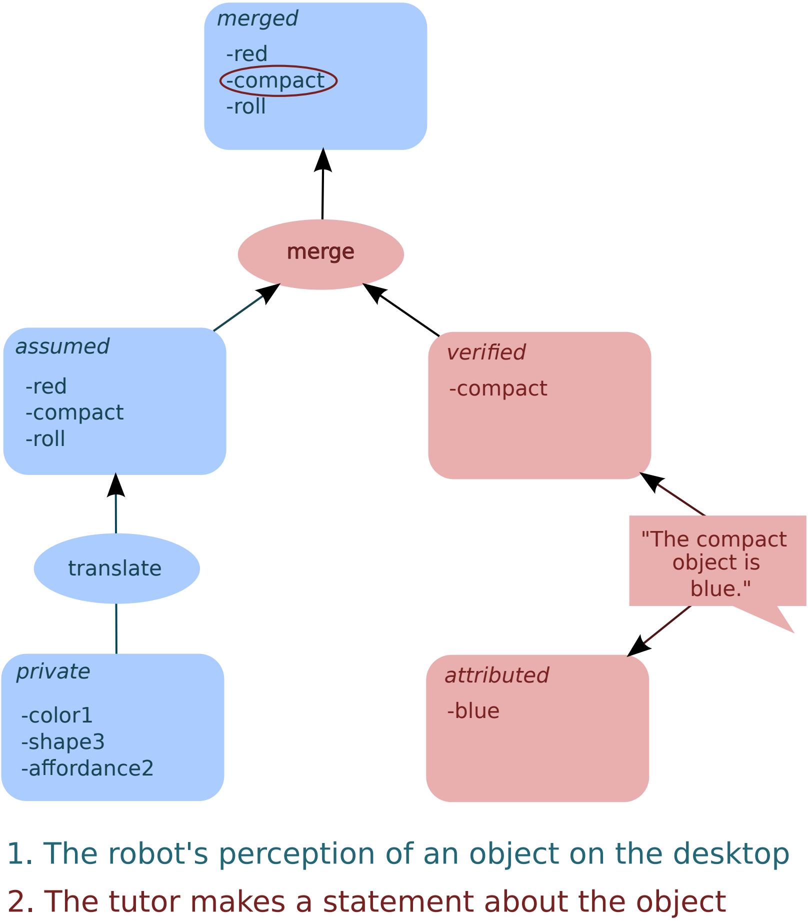
merged
-red
-compact
-roll
merge
assumed
-red
-compact
-roll
verified
-compact
translate
"The compact object is blue."
private
-color1
-shape3
-affordance2
attributed
-blue
1. The robot's perception of an object on the desktop
2. The tutor makes a statement about the object

Figure 4.3: A belief structure merging the robot's perception of an object with the description that a human tutor provides about the same object. At this stage, only the restrictive part of the description was merged. The red colour denotes the changes in the structure during this stage.

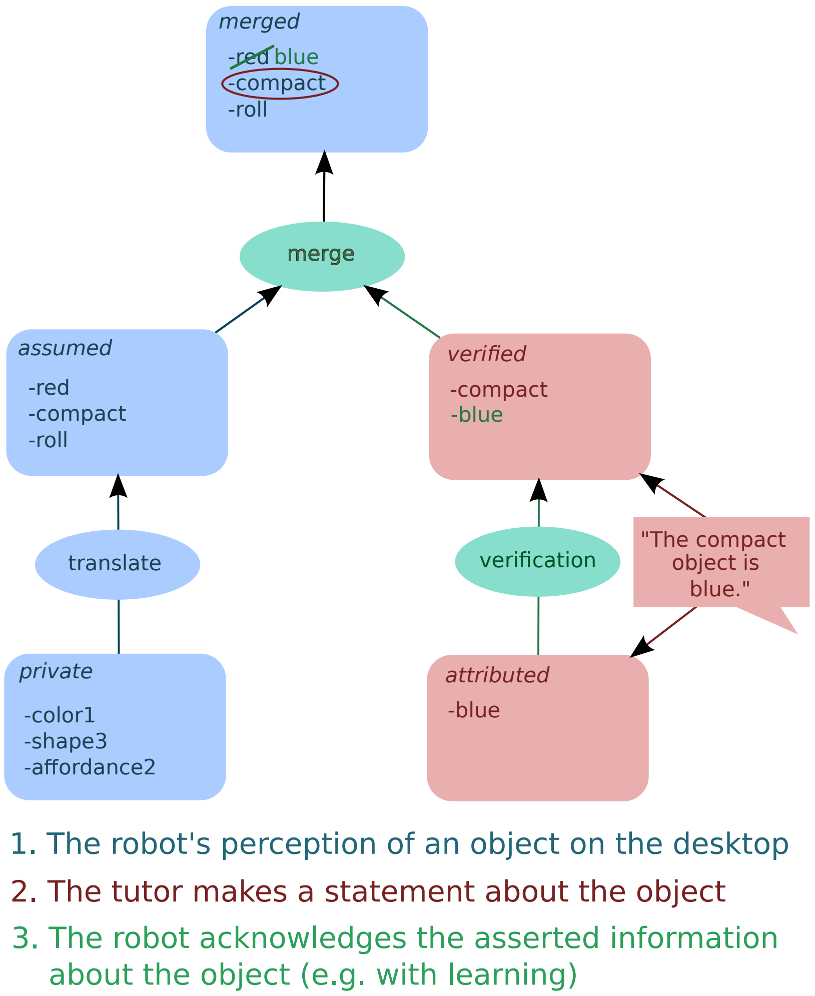
merged
-~~red~~ blue
-compact
-roll
merge
assumed
-red
-compact
-roll
verified
-compact
-blue
translate
verification
"The compact object is blue."
private
-color1
-shape3
-affordance2
attributed
-blue
1. The robot's perception of an object on the desktop
2. The tutor makes a statement about the object
3. The robot acknowledges the asserted information about the object (e.g. with learning)

Figure 4.4: The final belief structure that merges multi-agent information representing an object. After being verified, the assertive part of the tutor's utterance is merged into the structure, which creates a learning opportunity. The green colour denotes the changes in the structure during this stage.

4.3 Belief Based Cross-Modal Learning

One of the main purposes of George is to demonstrate certain learning paradigms that allow a cognitive system to improve its knowledge about its surroundings by exploiting information from different sources. In this process, the ability of the belief layer to manage, relate, and merge multi-modal and multi-agent information plays a vital role. From the belief standpoint, learning mechanisms exploit differences in information in beliefs pertaining to different modal or epistemological categories, but representing the same physical entity. Successful binding or reference resolution is therefore a key precondition for any kind of cross-modal learning.

At this point, it is necessary that we distinguish between two distinct types of cross-modal learning (the Encyclopedia of the Sciences of Learning distinguishes three such types [38]). In Chapter 2, we defined cross-modal learning as the process of improving the ability to merge (bind) multi-modal information. To implement such a process, a system needs special learners that learn how to associate concepts from different modalities. This kind of learning is described as *cross-modal learning on higher level of abstraction* in [38]. However, the same term may also refer to mechanisms exploiting multi-modal information for feeding modal learners, i.e. *weakly coupled cross-modal learning* according to [38]. We can apply the latter meaning to the learning mechanisms in George scenarios.

All learning paradigms that George implements try to obtain and use tutor's information about visible objects for learning visual concepts. In most cases, the tutor provides this information explicitly, as we saw in the example in Section 4.1.1. In these cases, the learning act is executed as a deliberate action issued by the planner, hence in a goal-driven fashion. We can categorize such learning mechanism as *explicit learning.* George's learning mechanisms fall into this category.

In contrast, *implicit learning* in principle completely bypasses planning and motivation, and occurs in a pure data-driven fashion (the difference between implicit and explicit learning is also explained in [45]). Implicit learning exploits the difference in information between assumed and merged beliefs to update modal visual concepts. Depending on the information that was merged, and the type of process that performs information merging (see Section 4.2), the difference can be (i) in the property confidence and (ii) in the property quality (e.g. as depicted in Figure 4.4). In the former case, the system has to have a difference threshold that triggers learning. In the latter case, the system can perform both, the learning of the right concept in the merged belief, and the unlearning of the wrong concept in the assumed belief. Of course, before the learning action takes place, the property information in the merged belief has to be translated back to modal symbols.

An important question when implementing implicit learning is when to trigger it. It would not be advisable simply to trigger the implicit learning after each merging, since this could result in relearning the same information several times. A better strategy is to compare the new merged belief with the old one, and react only when there is a change in the quality of merged information, or if confidence of the new information raises the confidence difference above the threshold.

Another problem concerning implicit learning occurs when the implicit learning is combined with the explicit learning. After the assertive information is used for learning, it is verified, and consequently merged into the merged belief. This can trigger the implicit learning, which means that the same information is used for learning twice. We can avoid this problem by simply restrict implicit learning to the restrictive information only (as is also the case in [45]). This means that the implicit learning is triggered after the first merging of the verified information, only. When used as a supplementary learning mechanism in combination with the explicit learning, it is important to adequately tune the effects of both learning mechanisms. The effect of implicit learning should be less pronounced, since it occurs more often in general, and not in a deliberate manner.

4.3.1 Learning Mechanisms in George

In this section, we describe the four mechanisms that govern the learning part of George's behaviour. They are all deliberate – trying to learn about properties explicitly. Save for situated autonomous learning, they are all based on the interaction with the tutor. They differ over which agent has the initiative in the interaction, and over whether the initiative stems from the extrospection (of the situation), or from introspection.

Situated Tutor-Driven Learning

The *situated tutor-driven learning* can be regarded as the most classic example of explicit learning. We refer to this case, when a tutor takes the initiative and explicitly tries to teach the robot something about the visible objects. There are two necessary conditions for such a learning act: (i) the visual subsystem detects an object and processes its visual features, and (ii) the information provided by the tutor is successfully attributed to the same object. This results in the creation of communicative intention containing both a reference to the object in question, and the inferred desired effect of the tutor's utterance (i.e., the corresponding change in the robot's private belief about the object). The intention structure is the prerequisite for the motivation subsystem to create a planning goal for visual learning. The goal will be committed to planning and execution only if the expected information gain for the learning action (provided by the visual subsystem) is high enough. Since both prerequisites for the learning are present (visual information from the private belief and a label from the intention), the planner generates a trivial plan – a sequence of learning actions, one for each property provided by the tutor. The execution subsystem triggers the visual learner in the Visual SA to update the internal visual models. This action also results in an updated *model status* belief, which maintains key meta-information about the visual models.

Situated Autonomous Learning

When a *merged belief* contains only the information provided by the vision subsystem, and this information is deemed reliable (the visual concept has been recognised with high a confidence), the motivation subsystem triggers an autonomous learning cycle. The models of the corresponding visual concepts are automatically updated, also resulting in an updated *model status* belief. In the case of a very confident recognition, such an update is not necessary because the current representation can describe the object perfectly well. However, when the recognition is slightly less reliable, it makes sense to adapt the knowledge to the perceived object, thus increasing the confidence of recognition of similar objects in the future. There is, however, a persistent danger of incorporating erroneously recognised information into the models in such an automated way; the system should therefore behave very conservatively, and only update the knowledge when the recognition is reliable enough, otherwise it should verify its decision by consulting the tutor.

Strictly speaking, *situated autonomous learning* does not categorize as cross-modal learning, since it involves information from visual modality, only. Since there is no explicit teaching intent from the tutor, it could be in principle implemented in a data-driven fashion, as implicit learning.

Situated Tutor-Assisted Learning

Depending on its current ability to recognise a specific object, George can ask the tutor a question about the object's properties. In this case, the merged belief that motivation acts upon contains only information from the private belief. To fully exploit its question, George asks about the object property with the highest *information gain* (as described in Section 3.1.3), expecting that the corresponding model would benefit most from the requested information. In the absence of attributed beliefs, the planner generates a plan to ask questions about missing information. The execution subsystem generates a corresponding intention, which the dialogue subsystem uses to synthesize a suitable utterance. Depending on confidence in the recognition results, the planner can choose between polar questions (that can be answered with 'yes' or 'no'), when recognition confidence is high (e.g. R: "Is the colour of the compact object red?"), and open questions (that require a label for the answer), when confidence is low (e.g. R: "What is the colour of the compact object?"). If the robot is unable to unambiguously verbally refer to the particular object, the planner can resort to the robot's arm. A common ground with the tutor is established by pointing at the object. Pointing reflects in the uttered question accordingly (e.g. R: “What is the colour this object?”). After the tutor answers, the workflow is similar to tutor-driven learning.

Non-Situated Tutor-Assisted Learning

In non-situated tutor-assisted learning paradigm, the robot tries to obtain new learning samples by making a request to the human tutor (e.g. R: “Could you show me something red?”). The robot relies on introspection to influence the quality of potential new objects. Introspection of property models is performed in the visual subsystem (Visual SA). The results are propagated to the belief layer in the form of epistemic structure *model status*, which contains key meta-information about the models maintained by the visual learner. The system can use the *information gain* to estimate the reliability of available models, without relating to any particular objects in the scene.

Chapter 5

Experiments

This chapter is divided in two parts. Section 5.1 describes the off-line experiments that were performed on a prototype binding system that was implemented according the principles described in Chapter 2. Section 5.2 describes the evaluation of the cognitive system described in Chapters 3 and 4.

5.1 Evaluation of the Binding Prototype

5.1.1 Experimental Setup

We implemented a prototype of our binding and cross-modal learning system (see Chapter 2) in Alchemy[1] [23]. Our experimental database comprehended three modalities: vision, language and affordance. The visual modality had 13 features in total: six for object colour, three for the general shape (compact, elongated, flat) and four for the geometric shape. Language had 13 features matching the visual features and eight features for object type (e.g. book, box, apple, etc.). The affordance modality had three features describing the possible outcomes of pushing an object. Overall, we had 54 fully featured object prototypes.

[1] Alchemy is a software package providing various inference and learning algorithms based on Markov logic networks.

We designed the learning samples to mimic the robot interaction with a human tutor (like in the George scenario), where the human was showing objects to the robot, describing their properties. The learning samples were organized in small batches. Each learning sequence consisted of 80 learning batches. We generated the batches randomly, with balanced appearances of object prototypes.

We designed 30 test-cases for evaluating the binding process. In each test-case, we had three visual percepts and one non-visual percept. The binder had to determine which visual percept, if any at all, the non-visual percept belonged to (i.e. four possible choices: one for each visual percept and one for no corresponding percept). Of the four possible choices, one was always more obvious than the others, and thus deemed correct. The possibility that the system inferred as the most probable was considered its binding choice, if the probability exceeded 30%. If the probability of the most probable choice was

$union = \{U1, U2, U3, U4\}$

$perFeat(P1, VRed), perFeat(P1, VFlat), perFeat(P1, VCylindrical),$

$perFeat(P2, VBlue), perFeat(P2, VCompact), perFeat(P2, VSpherical),$

$perFeat(P3, VGreen), perFeat(P3, VElongated), perFeat(P3, VConical),$

$uniPer(U1, P1), uniPer(U2, P2), uniPer(U3, P3)$

$perFeat(P4, LRed), perFeat(P4, LFlat), perFeat(P4, LCylindrical)$

$uniPer(u, P4)?$

Figure 5.1: An example of an easy test-case. We can see that objects represented with visual percepts ($P1$, $P2$ and $P3$) differ in all types of visual features. The system needs to determine which union the fourth, linguistic percept belongs to.

less than 30%, the case was automatically considered not correctly resolved.

The test-cases varied in their level of difficulty, and were divided in three categories:

- the *easy test-cases* featured distinct features for visual percepts and complete information for all percepts (each percept had a value for each feature type belonging to its modality, see Figure 5.1),
- the *medium test-cases* could have features shared by several percepts or incomplete percept information,
- the *hard test-cases* had both incomplete information and feature sharing (see Figure 5.2).

$$union = \{U1, U2, U3, U4\}$$

$$perFeat(P1, VRed), perFeat(P1, VCompact), perFeat(P1, VConical),$$

$$perFeat(P2, VGreen), perFeat(P2, VCompact), perFeat(P2, VSpherical),$$

$$perFeat(P3, VGreen), perFeat(P3, VFlat), perFeat(P3, VConical),$$

$$uniPer(U1, P1), uniPer(U2, P2), uniPer(U3, P3)$$

$$perFeat(P4, LApple)$$

$$uniPer(u, P4)?$$

Figure 5.2: An example of a difficult test-case. We can see that the objects represented with visual percepts ($P1$, $P2$ and $P3$) are less distinct than in the easier test-case (Figure 5.1), and with some incomplete information. The system has to find out which visual percept could be an apple. The visual training samples for apples consisted of compact and spherical percepts of red or green colour.

The tests were performed several times during the learning process, in intervals of four batches.

We performed our evaluation with three inference methods: *Belief propagation* [29, 30] and two Markov chain Monte Carlo (MCMC) sampling methods – *MC-SAT* [31] and *Gibbs sampling* [12]. In both MCMC methods, the number of sampling steps was limited to 2000 (with additional 100 burn-in samples), while the maximum number of iterations for Belief propagation was 2000.

5.1.2 Experimental Results and Evaluation

Figures 5.3 and 5.4 show the average rate of correct binding choices over 20 randomly generated learning sequences for all three inference methods. In all cases, the binding rate tends to grow and converge with the growing number of samples, though with some oscillations. The oscillations are more pronounced for the difficult test samples, especially in the case of the MC-SAT method. The MC-SAT method has also a lower correctness rate compared to Gibbs sampling and Belief propagation. This can be explained by the slower convergence rate per sampling step for the MC-SAT sampler, which is, however more than compensated by its speed (approximately ten times faster per step compared to the other two methods).

Analysing the results example by example, we identified several issues hindering the system performance. The subset of possible associations represented with the binding rules does not include *many-to-one feature associations* (e.g. *red*, *compact*, *cylindrical* ⇒ *colacan*). Such associations would be especially beneficial for situations reflected by certain difficult test-cases (see figure 5.2). Of course, to prevent combinatorial explosion, the addition of *many-to-one associations* would require quite a different (a much more selective) conceptual grounding strategy.

The feature types with less members are underestimated, e.g. both types of shapes with the respect to the colour type, which reflects the fact that is less likely for the same colour to appear in two or more percepts in the learning samples. This makes colour associations more distinctive than shape associations, and is in general in perfect accordance with [40]. In our case, however, there are situations, where this principle can represent a problem. We can see an example of such situation in figure 5.5, where a very distinctive colour association overweights the shape mismatch, which results in a wrong binding result.

A portion of test-cases (10%) represented situations where no existing visual percept matched the non-visual percept, which should have resulted in a separate percept union for the non-visual percept. In all test-cases of this kind the mismatching feature pairs outnumbered the matching ones in all potential two-percept unions (a scenario where, e.g. just one mismatching feature pair is enough to deem the percepts not compatible is difficult to formulate in pure probabilistic logic). In general, such situations are harder to resolve, since the system has to rely on segregative rules, only. A correct resolution requires the segregative associations to outweigh the aggregative

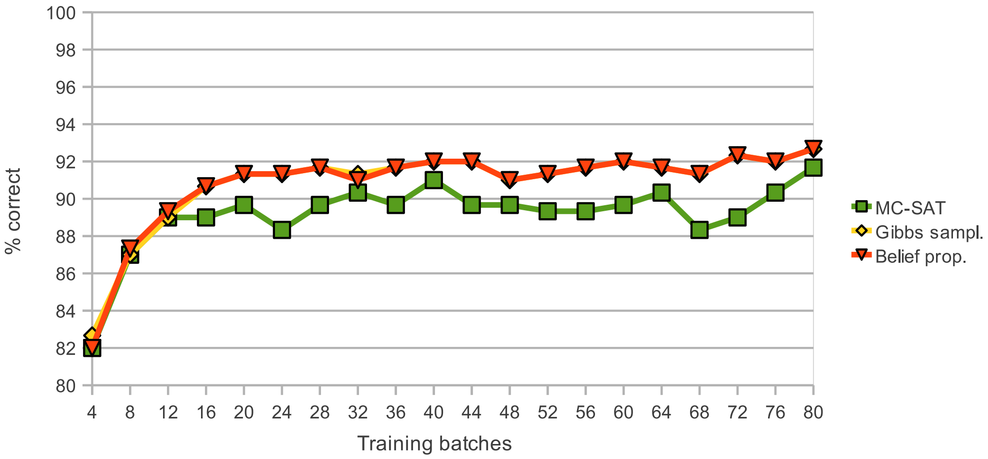

Figure 5.3: Experimental results: the average overall rate of correct binding choices relative to the number of training batches (10 randomly generated learning sequences were used). The green, yellow and red lines denote the three inference methods: MC-SAT, Belief propagation and Gibbs sampling, respectively.

associations in all plausible percept pair combinations. The correctness rate of these test-cases is lower and increases at a slower rate with more pronounced oscillations.

At this point, we have to emphasize again the off-line nature of these experiments. In this experimental setup, the system was forced to make a decision even in a very uncertain situation (e.g. in a situation where the probabilities of two most probable choices were very close). In contrast, in an integrated cognitive system, the binding mechanism would handle uncertain situations differently, e.g. by triggering a behaviour that would try to clarify the situation (like described in Section 3.1.7). This would, of course, involve other cognitive mechanisms. In this sense, such off-line evaluation can not show the real value of a cognitive mechanism.

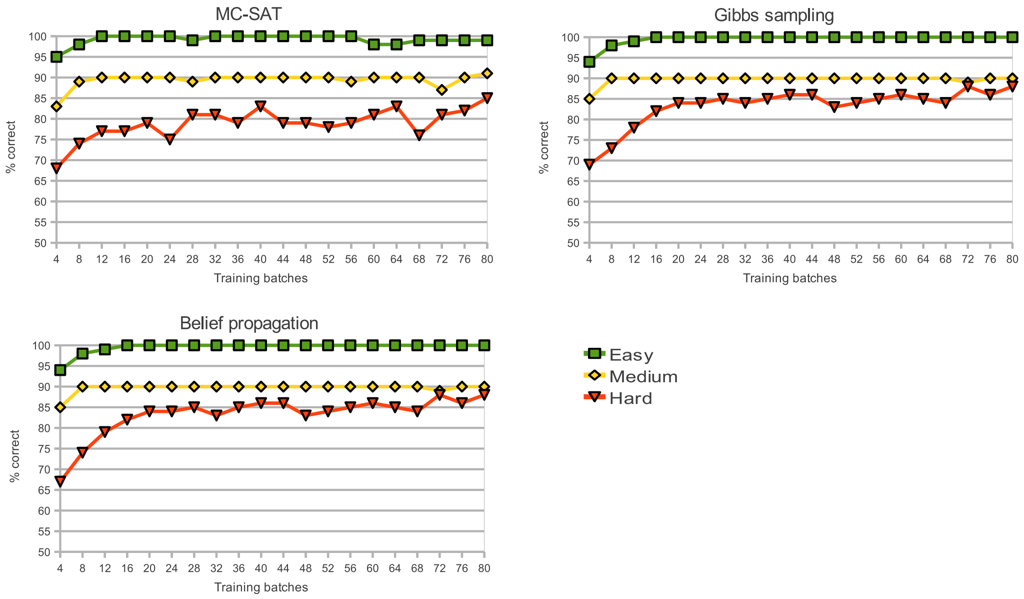

Figure 5.4: Experimental results by test-case difficulty: the average rate of correct binding choices relative to the number of training batches. The green, yellow and red lines denote the easy, medium and hard test samples, respectively.

$union = \{U1, U2, U3, U4\}$

$perFeat(P1, VRed), perFeat(P1, VCompact), perFeat(P1, VSpherical),$

$perFeat(P2, VRed), perFeat(P2, VElongated), perFeat(P2, VCylindrical),$

$perFeat(P3, VFlat), perFeat(P3, VCylindrical),$

$uniPer(U1, P1), uniPer(U2, P2), uniPer(U3, P3)$

$perFeat(P4, LRed), perFeat(P4, LFlat)$

$uniPer(U2, P4)$

Figure 5.5: An example of wrong binding. Because we have six possible colour values and only three for the general shape, the colour features are more distinctive. Hence, the colour associations have more impact on the binding process, which can sometimes result in wrong binding. In the case above, $U3$ is the correct union choice for percept $P4$. Instead, the system chooses $U2$ based on the red colour

5.2 Evaluation of George's Behaviour

The main goal of the experiments described in this section is to evaluate the behaviour of George in a real world environment, as well as the performance of the system as a whole. More specifically, we were interested in its interaction with its typical environment setting (i.e. the objects in its surroundings and a human tutor), with a particular emphasis on the mechanisms for interactive, situated learning. As these mechanisms depend on the ability of the system to merge multi-modal and multi-agent data (i.e. the belief layer), the experiments also represent the proof of concept for the approaches described in this work. It is not superfluous to emphasize at this point that the purpose of these on-line experiments was not the evaluation of individual modal recognisers and learners.

To illustrate the behaviour of the system during the learning process, we first present an example of interaction between a human tutor and the robot. Then we present the quantitative results, obtained by observing the robot's behaviour in a similar scenario.

5.2.1 An Example of Human-Robot Interaction

In this example of interaction, a human tutor and the robot engage in situated dialogue in order to improve robot's knowledge about visual concepts, such as colour, shape and object types. During the interaction, the robot aims to recognise and describe the objects on a table. The human can add, move or remove objects from the table, while teaching the robot about their properties. There can be up to five objects on the table at any time.

Initially, it is the tutor that has to drive the learning. But after a while, the robot can take the initiative, involving the tutor in his learning effort as he sees fit. Perhaps the most critical part of such interaction is establishing a common ground about the content of the scene. In each communicative act, the agents must explicitly or implicitly agree on which object they are talking about. Hence, the ability of merging multi-agent information is critical in such interaction.

At any time, the tutor can decide to ask questions about the objects in the scene, to see what the robot has learned so far. In this sense, the goal of the learning interaction is to achieve such a maturity of robot's representations that would make a correct description of the scene possible.

Let us consider an empty table. The tutor puts an object on the table. Applying its *attention mechanism*, the robot looks at it.

H: Do you know what this is?

R: No.

In the beginning, the robot knows nothing yet about any object. *Situated tutor-driven learning* is therefore imperative during these initial stages of the interaction, since the robot needs to initialise reliably its representations.

H: This is a red object.

R: Let me see. OK.

With this information, George can initiate its visual model of redness. After several similar learning steps, the acquired representations become reliable enough to allow George to reference verbally individual objects, and also understand references made by the human. This makes the whole interaction much easier. For example, the human can now ask situated questions even when there are more than one objects in the scene.

H: What colour is the coke can?

R: It is red.

When enough of the models are reliable, George can take the initiative, and drive the learning by asking questions himself. It will typically do this when he detects a new object in the scene, but can not reliably recognise all of its properties. In this case, the robot resorts to the *situated tutor-assisted* learning mechanism. In general, there are two possible kinds of gaps in robot's knowledge. If a property does not appear to fit any of the current models, the robot can asks the tutor to provide more information about the novel property with an open question:

R: What colour is this object?

H: It is yellow.

R: OK.

In the second case, the robot is able to associate the object property with a particular model, but the recognition is not very reliable. The robot can ask for clarification with a polar question:

R: Is this red?

H: No. This is yellow.

R: OK.

After receiving the answer, the robot corrects the representation of red (by unlearning the current sample from the model), and updates the representation of yellow.

If the recognition of an object property is somewhat more reliable, but still not perfectly reliable, George can update the models without asking any questions, employing the mechanism for *situated autonomous learning.* Since there is no external verification in this kind of learning behaviour, George tends to be very conservative, when using it.

The motivation subsystem makes George attend and process all the objects in the scene that were detected so far. When its knowledge about those objects is complete (i.e. knowledge about perceived instances, not necessarily knowledge on conceptual level[2]), it further explores the scene by looking around for new objects. If no additional objects are found, it resorts to introspection (the *non-situated tutor-assisted learning* mechanism) to detect gaps in its knowledge. It tries to attend potential knowledge gaps by asking the tutor to show an object with a particular property.

R: Please, show me something blue.

H puts an object at the table.

H: This is a blue object.

R: Thank you.

George prefers to refer to objects verbally, but it can do that also by pointing. Let us assume that the tutor empties the table, and then puts two new objects on the table. George detects and attends both. If it is able to reliably recognise and verbalise a distinctive set of attributes in both objects, it can refer to them verbally.

R: Is the Pepsi can blue?

H: Yes, it is.

R: Thank you.

However, if no verbal distinction is possible, the robot resorts to pointing.

R points at an object.

R: What colour is this object?

[2] By merging private and attributed information about objects, the robot is able to make sense of the scene. It is then able to talk and ask questions about the scene, even without understanding well the concepts that it uses to describe it. In the sense of Section 2.3, we could say that in the absence of own reliable knowledge, the robot is able to conceptually ground its representations to the knowledge of another agent.

H: It is yellow.

R: Thank you.

George uses the same method for generating references when requesting a clarification after an ambiguous question.

In such mixed-initiative dialogue, George continuously improves its representations of basic visual concepts. After a while, George can successfully recognise the acquired concepts to provide reliable answers to a variety of questions.

H: Do you know what is this?

R: It is a blue object.

H: What is the green object?

R: It is an ice tea bottle.

H: Which shape is it?

R: It is elongated.

5.2.2 Experimental Setup

It is very difficult to evaluate consistently such a complex, heterogeneous and asynchronous system. The system can exhibit different patterns of behaviour based on the visual input, and the timing and order of interactions, as well as the information provided by the tutor. To overcome this, we created a controlled experiment where we were able to vary the values of different variables, and systematically measure the performance of the system in terms of *achieved expected system behaviour*. We created an interaction scenario to invoke all of the different behaviours implemented in the system, involving different objects, placed on different positions. We ran this scenario ten times with the real robot, compared the resulting behaviour with the behaviour expected based on our design, and measured the rate of success. In this section we report the results, and analyse the system performance.

Scenario Setup

The scenario setup was similar to the one shown in Figure 3.1. We constrained the surface, where the tutor could place objects, to ten fixed locations across the table. These locations were unknown to the system. The area that the ten locations formed was wider than the camera view, hence the system had to use the pan-tilt unit to cover it.

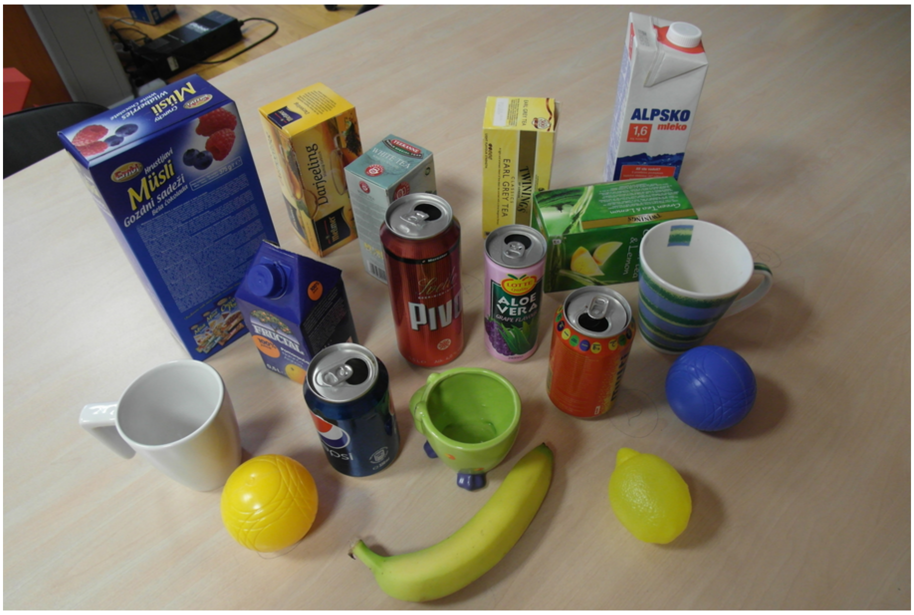

Figure 5.6: Objects used in the experiment.

We used 18 ordinary household objects. Each of them had one predominant colour (figure 5.6). We considered three concepts (colour, shape and type). Every iteration of the experiment was characterized by:

- Objects o_i: three objects selected among the objects depicted in Figure 5.6.
- Places p_i: three places selected among the ten predefined places, where the objects were positioned.
- Concepts $c^j \in \{colour, shape, type\}$.

- Concept values: $v_i^1 \in \{$*red, green, blue, yellow, ...*$\}$; $v_i^2 \in \{$*compact, elongated*$\}$, $v_i^3 \in \{$*milk box, banana, corn flakes, pepsi can, ...*$\}$,

where $i \in \{1, 2, 3\}$ is the index of the individual object, and $j \in \{1, 2, 3\}$ is the index of one of the concepts.

Actions

The experimental interaction consisted of a fixed sequence of actions (script) performed by the tutor. Table 5.1 presents the actions available to the tutor. During the interaction, the robot was expected to reply with the actions presented in Table 5.2. The scenario did not include all actions the robot was able to perform, nor all of the tutor's actions that were supported. Nevertheless, this set of actions could support all mechanisms of behaviour we intended to test, while it was sufficiently constrained to facilitate a consistent and controlled experiment.

Table 5.1: Tutor's actions in the experiment.

action	description and *example*
put(o,p)	Put the object o at the place p.
tellThis(v)	Tell the concept value v of the current object. *H: This is a red object.*
askValue(c,v)	Ask about the value of the concept c of the object referenced by another concept value v. *H: What shape is the yellow object?*
answerPolar	Answer a polar question. *H: Yes.*
answerOpen(v)	Answer an open question. *H: It is yellow.*

Table 5.2: A set of expected robot actions in the experiment.

action	description and *example*
attend(o)	Look at an object and analyse its properties.
askThisOpen(c)	Ask an open question about the current object. *R: What colour is this object?*
askThisPolar(v)	Ask a polar question about the current object. *R: Is this a mug?*
update(o,c,v)	Updates the model of the concept c with the value v using the features extracted from the object o.
lookAround	Looks around the scene.
askForObject(v)	Asks for an object with the concept value v. *R: Please, show me something green.*
answerValue(v)	Answers the question with the attribute value v. *R: It is a mug.*
askIfValue(v)	Verifies the referent using an attribute value v. *R: Do you mean the coffee box?*
point(o)	Points at an object o.
askIfPoint	Verifies the referent by pointing. *R: Do you mean this one?*

Interaction Script

Table 5.3 presents the interaction script that was used in the experiment. The script covers all the mechanisms of behaviour presented in Sections 3.3 and 4.3.1. Non-indented lines represent the tutor's actions, while the lines with expected robot actions are indented. We repeated this script ten times. For each script iteration, the object that we used, the locations that we put the object in, and the concepts that we discussed, were selected randomly. In other words, we varied the variables o, p, c, and v. However, in each iteration, we had to start with a suitable configuration of pre-learned property models that would allow the robot to act according to the script (e.g., a certain maturity of the property model is required to ask a modal question). As we have already pointed out, in these experiments, we are not concerned about the quality of the models the robot manages to build during the interaction.

Table 5.3: Scenario script.

1: put(o_1, p_1), put(o_2, p_2),	20: askValue(c, v) not requiring
p_1 and p_2 are far apart	disambiguation
2: attend(o_1)	21: answerValue(v)
3: analyseAsk(o_1)	22: askValue(c, v) requiring
4: answer(o_1, c, v)	verbal disambiguation
5: update(o_1, c, v)	23: askIfValue(v)
6: lookAround	24: answerPolar
7: attend(o_2)	25: answerValue(v)
8: analyseAsk(o_2)	26: askValue(c, v) requiring
9: answer(o_2, c, v)	disambiguation by pointing
10: update(o_2, c, v)	27: point(o)
11: lookAround	28: askIfPoint
12: askForObject(v)	29: answerPolar
13: put(o_3, p_3)	30: answerValue(v)
14: attend(o_3)	
15: tellThis(v)	where:
16: update(o_3, c, v)	analyseAsk(o):={askThisOpen(c) $\vee$
17: analyseAsk(o_3)	askThisPolar(v) $\vee$ /}
18: answer(o_3, c, v)	answer(o, c, v):={answerOpen(v) $\vee$
19: update(o_3, c, v)	answerPolar $\vee$ / $\vee$ tellThis(v)}

Actions analyseAsk(o) and answer(o, c, v) in Table 5.3 have multiple possibilities ('/' means 'do nothing'). The robot would choose its reaction to the current observation based on its reliability (e.g. what kind of question to ask, or whether to simply update the knowledge). When reacting to a question, the tutor would also tune his choice to the context. Either by simply answering the question, or by ignoring it and perhaps explicitly providing the desired information (e.g. when reacting to a polar question).

Each session begins with tutor placing two objects on the table. They are positioned sufficiently apart, so that only one of them is in the current camera view. The robot first analyses the visible object. Depending on results of the analysis, it may update the knowledge autonomously, or request some additional information from the tutor. After updating the models, the robot looks around in search for more objects. When it finds the second object, it attends to this object in a similar way. After the robot observes that there are no other objects on the tabletop, it asks for a new one, possibly with the property that it is currently most interested in. After the tutor complies, the robot again attends to the new object in a similar fashion.

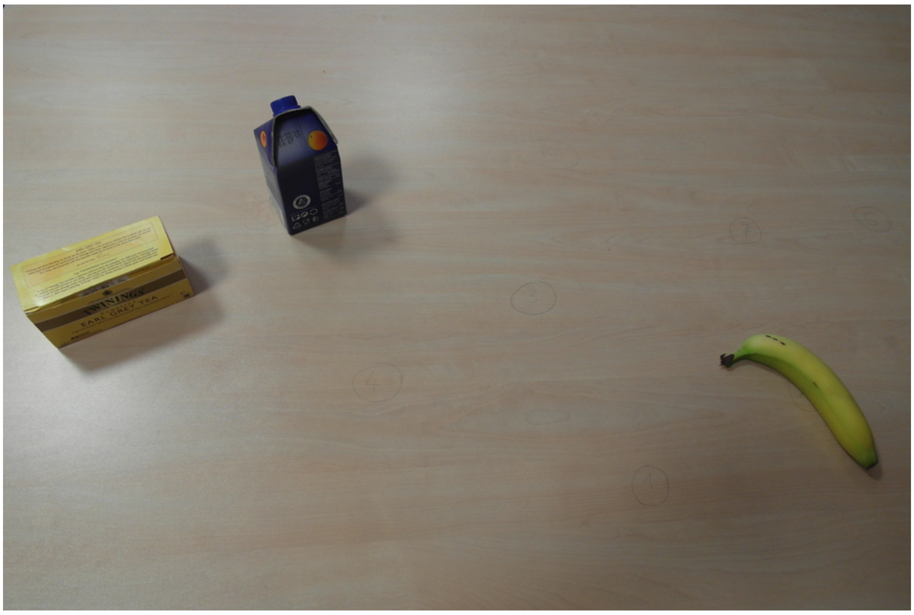

Figure 5.7: A typical scene from the robot's viewpoint.

At the end of each session, the tutor verifies the robot's knowledge by asking three questions about the objects on the table. The first question is unambiguous, and the robot is expected to answer immediately. For example, let us consider the scene depicted in Figure 5.7. If the robot is able to recognise the colours of the objects, this question might be H: "What shape is the blue object?". The second question is ambiguous, but it can be disambiguated by referring to another object property (e.g. H: "What shape is the yellow object?", R: "Do you mean the tea box?"). In the third case, the disambiguation can only be performed by pointing (e.g., H: "What is the yellow object?", R:"Do you mean this one?"). In all three cases, the robot is expected to perform the adequate actions to answer the question.

5.2.3 Experimental Results and Evaluation

Table 5.4 presents the results of the experiments. The results are grouped by mechanisms of behaviour (see Sections 3.3 and 4.3.1). For each mechanism, the table lists the *lines* from the script (Table 5.3) implementing it. In the last two columns, we can see how many times the specific actions were expected to be triggered (#exp), and how many times these actions were actually successfully executed (#exec). We evaluated the system performance by comparing the numbers in both columns.

Table 5.4: Experimental results - expected and executed actions.

mechanism	lines	#exp.	#exec.
Attention mechanism	2;7;14	30	30
Situated tutor-driven learning	5;10;16;19	18	18
Situated autonomous learning	3,5;8,10;17,19	9	9
Situated tutor-assisted learning	3,5;8,10;17,19	32	31
Exploring the scene	6;11	20	20
Non-sit. tutor-assisted learning	12	16	16
Answering tutor's requests 1	21	10	10
Answering tutor's requests 2	23,25	10	10
Answering tutor's requests 3	27,28,30	10	7

The performance of all learning mechanisms was almost impeccable, with only one learning failure in 75 cases. The system also exhibited a good performance for its other mechanisms of behaviour. The attention mechanism was triggered whenever expected. The detection of the objects was also very reliable (in the sense that it was correct when expected so). The system explored the scene whenever it was necessary.

Most of the tutor's questions were answered as expected, especially when no disambiguation was necessary, or when the robot could disambiguate the question verbally. The only problematic mechanisms was the disambiguation with pointing. On two occasions, the execution of the pointing action failed (along with the subsequent retries). Although the arm did point at the object, the execution mechanism was not able to report the execution completion and success to the planner. In one iteration, instead of pointing, the robot tried to disambiguate by the same property type that had been the object of the question (e.g. H: "What colour is the mug?", R: "Do you mean the red one?"). In a normal conversation, this could have been even considered appropriate, e.g. as a form of tentative answer. In our case, we took it as a failure, since the system had not exhibited the expected behaviour. George is actually designed to give tentative answers, but in different forms (e.g. "It might be red."), and under different circumstances (e.g. when it is not sure about the model).

Further analysing the mechanisms that rely on merged information, we could see that the data-merging processes in the belief layer worked as expected for all evaluated mechanisms, in all iterations, even when the outcome of the whole mechanism was not as expected.

In general, we can conclude that the system mostly exhibited the expected behaviour, and the observed failures were due to undiagnosed problems in our software, rather than problems with principles underlying our approach.

Chapter 6

Conclusion

In this thesis, we addressed a critical problem of any cognitive architecture aiming to operate in a realistic environment – the problem of bridging the *semantic gap* between lower, *multi-modal* cognitive layers, and higher, *a-modal* cognition. To bridge the semantic gap, a cognitive system has to be able to relate and merge information from different sources, to produce unified representations that can be used by higher cognitive processes. In order to make this process more flexible, the system must also include mechanisms for adapting and improving the *cross-modal knowledge* that is used for merging information.

We approached this problem by first developing a theoretical model of *binding and cross-modal learning.* We assumed an open and uncertain environment, where the system has to cope continuously with uncertainty and novelty in its perceptions. This implied a probabilistic approach to our modelling. We based our problem definition on Agrawal's problem of *association rule learning*, which we extended with the notions of *modalities*, *percepts* and *percept unions.* We described binding as the optimization of mapping a *percept configuration* to a possible *union configuration* based on accumulated cross-modal knowledge. By these definitions, we formulated a probabilistic binding mechanism and a cross-modal learner in Markov Logic Networks. We discussed a possible way of integrating such a mechanism into a cognitive architecture.

Another aim of our research to was to develop an approach to integration of our principles and methods into a real cognitive system. To this end, we have co-developed George, a prototype robot designed to continuously learn about its environment in a situated dialogue with a human tutor. George is based on a distributed asynchronous architecture, which facilitates a meaningful integration of several components that implement various cognitive processes. This results in a coherent system, capable of meaningful behaviour. Our instantiation of this architecture, i.e. George, focuses on several mechanisms of behaviour that facilitate interactive learning.

Instrumental for the learning mechanisms, but also crucial to the cognitive system in general, is the *belief layer*. The belief layer merges multi-modal and multi-agent information into unified representations that can be used by higher cognitive processes. The belief layer in George represents our exercise in integration of the principles of binding and cross-modal learning into a real cognitive system. In fact, though not modelled exactly as cross-modal binding, the belief layer incorporates most of its principles. At the same time, it also expands upon them by including methods of managing multi-agent information. An important association mechanism (akin to binding) in the belief layer is *reference resolution*. Reference resolution relates robot's own perceptions of a physical entity to the human description of the same entity.

We validated our approach with two sets of experiments. First, we evaluated in an off-line fashion a prototype binding system on our experimental database. The results show how the binding ability of the system increases with the number of samples, and how all this relates to the difficulty of the binding tasks. The results also point out some specific problems of the method that need to be addressed in the future. However, it is important to stress again that these were off-line experiments, where the system was forced to make a decision even in an uncertain situation. When part of an integrated cognitive system, many of such situations would be handled differently, e.g. by resorting to a clarifying behaviour that employs several additional cognitive mechanisms.

The on-line experiments performed on the George prototype represent the second part of validation of our approach – evaluation as a part of a real cognitive architecture. The focus of these experiments was on the evaluation of George's behaviour, with an emphasis on the mechanisms for interactive, situated learning. The experiments took place in a controlled real world environment, where George and a human tutor engaged in a situated dialogue, according to an interaction scenario. We evaluated the performance by comparing the actual behaviour of the robot to the expected behaviour (from the scenario). The results confirmed the ability of the system to actively pursue knowledge in a situated dialogue, with all of its learning mechanisms. The results also showed a correct performance of the belief layer, which functioned as expected in all situations.

We can conclude that the experiments confirmed the validity of our approach, both off-line and as an integral part of a cognitive architecture. Of course, despite all its architectural and technical complexity, George is still a very simple cognitive system in terms of supported behavioural and perceptual capabilities. It features one real perceptual modality, only. We 'borrowed' the second modality from the dialogue subsystem, though the relation between both information sources can be described more appropriately as multi-agent. A cognitive system with two or more perceptual modalities would surely represent an additional challenge in integration efforts, but also a great opportunity for further validation of our approach. In this sense, we perhaps missed an opportunity to treat the attentive part of George's vision as a separate modality[1]. In a similar fashion, multiple foreign agents would increase the challenge for the belief layer, especially in the epistemological sense.

Another possible future task could be to explore the possibilities of ex-

[1] In the present case, a sort of data merging from both visual sources actually occurs, even in a deliberate fashion, involving motivation and planning to move the camera, but within a single representation or percept, using the location of the item as the only clue.

tending the structure of cross-modal knowledge database with more complex rules. Of course, to accommodate more complex rules, we would also require more sophisticated learning methods. The present cognitive system is far from exploiting the full potential of MLN, and it is our firm opinion that MLN has a great potential for probabilistic cognitive modelling. Hence, a path worth pursuing might be to involve MLN in modelling and integration of other cognitive processes, as well.

Appendices

Appendix A

Povzetek magistrske naloge v slovenskem jeziku

A.1 Uvod

Spoznavne sisteme lahko najučinkoviteje opišemo kot sisteme, ki na podlagi razumevanja informacij sprejemajo premišljene odločitve. To dosežejo z organiziranem izvajanjem spoznavnih operacij, kot so prepoznavanje, analiziranje, povezovanje, odločanje, načrtovanje, itd. Za umetne spoznave sisteme, ki delujejo v realnem okolju, je torej nujna sposobnost zbiranja in razumevanja relevantnih informacij o svoji okolici, podlagi katerih se lahko lahko samostojno odločajo ali načrtujejo svoje nadaljnje dejavnosti. V splošnem lahko spoznavni sistemi zbirajo informacije o okolici na dva načina: (i) z interpretacijo podatkov iz senzorjev, oziroma s *percepcijo*, ali (ii) z interpretacijo podatkov drugega agenta, če je sistem sposoben *komunikacije* z njim. Percepcija je seveda bolj neposreden in učinkovitejši od obeh načinov. Vendar pa za uspešno percepcijo sistem potrebuje ustrezno konceptualno znanje, ki ga mora tudi nadgrajevati, če deluje v odprtem in dinamičnem okolju. Če ima sistem več različnih senzorjev in več podsistemov, ki interpretirajo senzorske podatke, govorimo o *večmodalnosti.* V tem primeru za delovanje potrebuje tudi *čezmodalno znanje*, s katerim povezuje informacije, ki izhajajo iz različnih tipov zaznav. Tovrstno znanje dopolnjuje s *čezmodalnim učenjem.* Na ta način sistem tudi premošča semantični prepad med nižjenivojskimi in višjenivojskimi spoznavnimi procesi.

Spoznavni sistem lahko s svojim vedenjem načrtno pripomore k učinkovitejšem učenju. Če se v njegovi okolici nahaja človek, s katerim lahko komunicira, predstavlja to priložnost, da s pogovorom hitreje dopolni svoje dojemanje (oz. percepcijo) okolice, kot tudi svoje konceptualno znanje. Seveda pa vedenjski mehanizmi za interaktivno učenje še povečujejo kompleksnost podsistemov za večmodalno združevanje informacij.

A.2 Čezmodalno povezovanje in učenje

V drugem poglavju opisujemo teoretični model čezmodalnega povezovanja in učenja. Predpostavili smo odprto in nepredvidljivo okolje, kjer mora sistem biti kos negotovim percepcijam in novim konceptom. Zato smo se odločili za verjetnostno modeliranje. Najprej smo definirali problem. Za osnovo smo vzeli *Agrawalov problem učenja asociativnih pravil*, ki smo ga razširili s pojmom modalnosti. Vsaka modalnost prispeva svojo percepcijo elementov v okolici, sistem pa mora percepcije pravilno združiti v enoten opis okolja. Če so percepcije sestavljene iz več elementov, je mogočih več opisov. Čezmodalno povezovanje smo definirali kot iskanje optimalnega opisa na podlagi čezmodalnega znanja. Na podlagi te definicije smo formulirali mehanizem verjetnostnega čezmodalnega povezovanja in učenja v *markovskih logičnih omrežjih*. Poglavje zaključujemo z razpravo o integraciji takega mehanizma v spoznavno arhitekturo.

A.3 Prototip spoznavnega sistema George

Pristop iz drugega poglavja smo hoteli ovrednotiti kot del delujočega spoznavnega sistema. V tretjem poglavjiu opisujemo prototip takega sistema, ki smo ga v sodelovanju s partnerji razvili prav z namenom preizkušanja skupnega delovanja različnih spoznavnih mehanizmov. George je inteligenten robot, ki z opazovanjem predmetov v svoji okolici in s pomočjo pogovora s človekom nenehno dopolnjuje svoje konceptualno znanje o lastnostih predmetov. Osnovo sistema predstavlja distribuirana arhitekturna shema (Cognitive Architecture Schema), ki z asinhrono integracijo spoznavnih komponent omogoča razvoj koherentnih spoznavnih arhitektur. Naša udejanitev te arhitekturne sheme, torej George, se osredotoča na vedenjske mehanizme, ki omogočajo interaktivno učenje.

A.4 Premoščanje semantičnega prepada

V tem poglavju podrobneje opišemo *podsistem prepričanj*, ki je pomemben del spoznavnega sistema George, saj premošča semantični prepad med njegovimni modalnimi in nemodalnimi spoznavnimi procesi. Prav tako je to ključni del večine njegovih vedenjskih mehanizmov, predvsem tistih za interaktivno učenje, hkrati pa predstavlja tudi materializacijo naših načel integracije čezmodalnega povezovanja v spoznavne sisteme. Podsistem prepričanj načela čezmodalnega povezovanja nekoliko prilagodi, predvsem pa nadgradi z nekaterimi epistemičnimi načeli za obdelavo in združevanje večagentne informacije. V tem smislu je pomemben mehanizem *določanja sklicevanja*. Gre za mehanizem podoben čezmodalnemu povezovanju, ki združuje robotovo lastno percepcijo elementov okolice z ustreznim opisom, ki ga poda sogovornik (npr. človek). Na koncu poglavja v dani kontekst vključimo diskusijo o načinih čezmodalnega učenja in podrobneje opišemo Georgove vedenjske mehanizme za interaktivno učenje.

A.5 Eksperimenti

Naš pristop k večmodalnem združevanju informacije smo ovrednotili z dvema vrstama experimentov. Najprej smo z eksperimenti, ki simulirajo trimodalno spoznavno arhitekturo, samostojno ovrednotili prototip mehanizma za čezmodalno povezovanje in učenje. Testne naloge smo razdelili v tri skupine: lažje, srednje in težje. Rezultati so pokazali, kako sposobnost čezmodalnega povezovanja narašča s številom učnih primerov in kako se na tem odraža težavnost testnih nalog. Nakazali so tudi nekatere specifične probleme metode, ki bi jih bilo dobro obravnavati v prihodnosti. Pri tem pa gre še enkrat poudariti, da gre za samostojne eksperimente, kjer se je bil mehanizem primoran odločiti tudi v zelo negotovih situacijah. Če bi bil del koherentnega spoznavnega sistema, bi le-ta velikokrat v tovrstnih razmerah ravnal bolj celostno. Lahko bi npr. prepustil odločitev drugemu spoznavnemu mehanizmu ali sprožil aktivnosti za razjasnitev okoliščin, ki vključujejo več njih.

Drugi del poglavja opisuje eksperimente, ki smo jih izvedli na prototipu robota George, s ciljem vrednotenja njegovih vedenjskih mehanizmov (predvsem mehanizmov za interaktivno učenje). Na ta način smo ovrednotili tudi naš pristop k večmodalnem združevanju informacij kot del delujoče spoznavne arhitekture. Eksperimenti so potekali v nadzorovanem realnem okolju, kjer sta se robot in njegov učitelj pogovarjala o predmetih v okolici po vnaprej določenem scenariju. Robotovo obnašanje smo ovrednotili s primerjavo njegovega vedenja s pričakovanim vedenjem v scenariju. Rezultati so potrdili pravilno delovanje vseh vedenjskih mehanizmov robota in s tem tudi njegovo sposobnost interaktivnega učenja. Pri tem je podsistem prepričanj deloval v skladu s pričakovanji v vseh situacijah.

A.6 Zaključek

Z rezultati eksperimentov smo torej potrdili perspektivnost našega pristopa pri premoščanju semantičnega prepada v spoznavnih sistemih. Velja pa poudariti, da je George, navkljub tehnični in arhitekturni kompleksnosti, zelo preprost spoznavni sistem, kar se tiče njegovih sposobnosti percepcije in vedenja. Ima zgolj eno pravo modalnost, tako da smo si morali kot drugo modalnost 'sposoditi' podsistem za dialog. V tem smislu bi spoznavni sistem z več pravimi modalnostmi predstavljal dober izziv za prihodnost.

Podobno velja za okolje z več kot enim sogovornikom. Veliko rezerv in s tem možnosti za delo v prihodnosti vidimo v formulaciji kompleksnejših modelov v markovskih logičnih omrežjih, pa tudi pri njihovi uporabi v drugih spoznavnih procesih.

Bibliography

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. *Proc. of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C.*, pages 207–216, May 1993.

[2] A. Bartels and S. Zeki. The temporal order of binding visual attributes. *Vision research*, 46(14):2280–2286, 2006.

[3] J. Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society. Series D (The Statistician),*, 24(3):179–195, 1975.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 23(11):1222–1239, 2001.

[5] M. Bratman. *Intentions, Plans, and Practical Reason.* Harvard University Press, Cambridge, MA, USA, 1987.

[6] M. Brenner and B. Nebel. Continual planning and acting in dynamic multiagent environments. *JAAMAS*, 19(3):297–331, 2009.

[7] A. Chella, M. Frixione, and S. Gaglio. A cognitive architecture for artificial vision. *Artif. Intell.*, 89(1-2):73–111, 1997.

[8] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, March 1990.

[9] P. Domingos. Toward knowledge-rich data mining. *Data Min. Knowl. Discov.*, 15:21–28, August 2007.

[10] P. Domingos and M. Richardson. Markov logic: A unifying framework for statistical relational learning. In *Proc. of the ICML-2004 workshop on statistical relational learning and Iits connections to other fields*, pages 49–54, 2004.

[11] K. T. Fann. *Peirce's Theory of Abduction.* Mouton, The Hague, The Netherlands, 1970.

[12] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 6:721–741, 1984.

[13] J. J. Gibson. The theory of affordances. *Perceiving, Acting, and Knowing*, Towards an Ecological Psychology:127–143, 1977.

[14] W. R. Gilks and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice.* Chapman & Hall/CRC, 1996.

[15] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42:335–346, 1990.

[16] N. Hawes. A survey of motivation frameworks for intelligent systems. *Artificial Intelligence*, 175(5-6):1020–1036, 2011.

[17] N. Hawes and J. Wyatt. Engineering intelligent information-processing systems with CAST. *Adv. Eng. Inform.*, 24(1):27–39, 2010.

[18] N. Hawes, J. Wyatt, M. Sridharan, H. Jacobsson, R. Dearden, A. Sloman, and G.-J. Kruijff. *Architecture and Representations*, volume 8 of *Cognitive Systems Monographs*, pages 51–93. Springer Berlin Heidelberg, April 2010.

[19] M. Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[20] H. Jacobsson, N. Hawes, G-J. Kruijff, and J. Wyatt. Crossmodal content binding in information-processing architectures. In *Proc. of the 3rd ACM/IEEE International Conference on Human-Robot Interaction*, Amsterdam, March 2008.

[21] H. Jacobsson, N. Hawes, D. Skočaj, and G-J. Kruijff. Interactive learning and cross-modal binding - a combined approach. In *Symposium on Language and Robots*, Aveiro, Portugal, 2007.

[22] M. Janíček. Abductive reasoning for continual dialogue understanding. In Marija Slavkovik and Daniel Lassiter, editors, *New Directions in Logic, Language, and Computation*. Springer, 2012.

[23] S. Kok, M. Sumner, M. Richardson, P. Singla, H. Poon, D. Lowd, J. Wang, and P. Domingos. The alchemy system for statistical relational ai. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2009.

[24] M. Kopicki. *Prediction learning in robotic manipulation*. PhD thesis, University of Birmingham, 2010.

[25] M. Kristan and A. Leonardis. Online discriminative kernel density estimator with gaussian kernels. *IEEE Trans. Syst. Man Cybern. B: Cybernetics*, 2013.

[26] M. Kristan, D. Skočaj, and A. Leonardis. Online Kernel Density Estimation for interactive learning. *Image and Vision Computing*, 28(7):1106–1116, July 2010.

[27] P. Lison, C. Ehrler, and G.-J. Kruijff. Belief modelling for situation awareness in human-robot interaction. In *Proceedings of the 19th IEEE International Symposium in Robot and Human Interactive Communication*. IEEE, 2010.

[28] F. Niu, C. Ré, A. Doan, and J. W. Shavlik. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *In Proc. of the 37th International Conference on Very Large Data Bases*, 4(6):373–384, 2011.

[29] J. Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Proc. of the 2nd National Conference on Artificial Intelligence*, pages 133–136. AAAI Press, 1982.

[30] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[31] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, pages 458–463. AAAI Press, 2006.

[32] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

[33] D. Roth. On the hardness of approximate reasoning. *Artif. Intell.*, 82(1-2):273–302, 1996.

[34] D. Roy. Learning visually-grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3-4):353–385, 2002.

[35] D. Roy. Grounding words in perception and action: computational insights. *TRENDS in Cognitive Sciences*, 9(8):389–396, 2005.

[36] W. Singer. Consciousness and the binding problem. *Annals of the New York Academy of Sciences*, 929:123–46, 2001.

[37] D. Skočaj, M. Kristan, A. Vrečko, M. Mahnič, M. Janiček, G.-J. M. Kruijff, M. Hanheide, N. Hawes, T. Keller, M. Zillich, and K. Zhou. A system for interactive learning in dialogue with a tutor. In *IEEE/RSJ*

International Conference on Intelligent Robots and Systems IROS 2011, 2011.

[38] D. Skočaj, G.-J. Kruijff, and A. Leonardis. Cross-modal learning. In *Encyclopedia of the Sciences of Learning*, pages 861–864. Springer, 2012.

[39] D. Skočaj, A. Vrečko, M. Mahnič, M. Janicek, G.-J. Kruijff, M. Hanheide, N. Hawes, J. Wyatt, T. Keller, K. Zhou, M. Zillich, and M. Kristan. An integrated system for interactive continuous learning of categorical knowledge. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, o.A.:o.A., 2015.

[40] L. Steels. *The Talking Heads Experiment. Volume 1. Words and Meanings.* Laboratorium, Antwerpen, 1999.

[41] M. Stone and R. H. Thomason. Coordinating understanding and generation in an abductive approach to interpretation. In *Proceedings of DIABRUCK 2003*, 2003.

[42] D. Vernon, G. Metta, and G. Sandini. A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *Evolutionary Computation, IEEE Transactions on*, 11(2):151–180, April 2007.

[43] A. Vrečko, M. Janíček, A. Leonardis, and D. Skočaj. Associating and merging multi-modal and multi-agent information in a cognitive system. Technical Report TR-LUVSS-02/2012, University of Ljubljana, Faculty of Compuer and Information Science, 2012.

[44] A. Vrečko, A. Leonardis, and D. Skočaj. Modeling binding and cross-modal learning in markov logic networks. *Neurocomputing*, 96:29–36, November 2012.

[45] A. Vrečko, D. Skočaj, N. Hawes, and A. Leonardis. A computer vision integration model for a multi-modal cognitive system. In *Proc. of the*

2009 IEEE/RSJ Int. Conf. on Intelligent RObots and Systems, pages 3140–3147, St. Louis, Oct. 2009.

[46] J. Wyatt, A. Aydemir, M. Brenner, M. Hanheide, N. Hawes, P. Jensfelt, M. Kristan, G-J. Kruijff, P. Lison, A. Pronobis, K. Sjöö, D. Skočaj, A. Vrečko, H. Zender, and M. Zillich. Self-understanding & self-extension: A systems and representational approach, 2010. Accepted for publication.

[47] J. Wyatt, A. Aydemir, M. Brenner, M. Hanheide, N. Hawes, P. Jensfelt, M. Kristan, G.-J. Kruijff, P. Lison, A. Pronobis, K. Sjöö, A. Vrečko, H. Zender, M. Zillich, and D. Skočaj. Self-understanding and self-extension: A systems and representational approach. *IEEE TAMD*, 2(4):282 – 303, December 2010.

[48] J. S. Yedidia, W. T. Freeman, and Y. Weiss. *Understanding belief propagation and its generalizations.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[49] K. Zhou, A. Richtsfeld, M. Zillich, and M. Vincze. Coherent spatial abstraction and stereo line detection for robotic visual attention. In *Proceedings of IROS 2011*, 2011.

[50] K. Zhou, K. M. Varadarajan, M. Zillich, and M. Vincze. Web mining driven semantic scene understanding and object localization. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Phuket, Thailand, Dec 2011.

[51] M. Zillich, J. Prankl, T. Mörwald, and M. Vincze. Knowing your limits - self-evaluation and prediction in object recognition. In *Proceedings of IROS 2011*, 2011.