An integrated system for interactive continuous learning of categorical knowledge

Skocaj, Danijel; Vrecko, Alen; Mahnic, Marko; Janícek, Miroslav; Kruijff, Geert-Jan M; Hanheide, Marc; Hawes, Nick; Wyatt, Jeremy L; Keller, Thomas; Zhou, Kai; Zillich, Michael; Kristan, Matej

DOI:
10.1080/0952813X.2015.1132268

To appear in the *Journal of Experimental & Theoretical Artificial Intelligence*
Vol. 00, No. 00, Month 20XX, 1–29

An integrated system for interactive continuous learning of categorical knowledge

Danijel Skočaj[a*], Alen Vrečko[a], Marko Mahnič[a], Miroslav Janíček[b], Geert-Jan M Kruijff[b], Marc Hanheide[c], Nick Hawes[c], Jeremy L Wyatt[c], Thomas Keller[d], Kai Zhou[e], Michael Zillich[e] and Matej Kristan[a]

[a] *University of Ljubljana, Slovenia;* [b] *DFKI, Saarbrücken, Germany;* [c] *University of Birmingham, UK;* [d] *Albert-Ludwigs-Universität, Freiburg, Germany;* [e] *Vienna University of Technology, Austria*

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

This article presents an integrated robot system capable of interactive learning in dialogue with a human. Such a system needs to have several competencies and must be able to process different types of representations. In this article we describe a collection of mechanisms that enable integration of heterogeneous competencies in a principled way. Central to our design is the creation of beliefs from visual and linguistic information, and the use of these beliefs for planning system behaviour to satisfy internal drives. The system is able to detect gaps in its knowledge and to plan and execute actions that provide information needed to fill these gaps. We propose a hierarchy of mechanisms which are capable of engaging in different kinds of learning interactions, e.g. those initiated by a tutor or by the system itself. We present the theory these mechanisms are build upon and an instantiation of this theory in the form of an integrated robot system. We demonstrate the operation of the system in the case of learning conceptual models of objects and their visual properties.

Keywords: Cognitive system, interactive learning, motive management, knowledge gap detection, extrospection, introspection

1. Introduction

Cognitive systems are characterised by their ability to learn, communicate, and act autonomously. These capabilities are clearly required in a robot that is supposed to collaborate with a human in a real-world environment. Such a system should be able to communicate with its companion, autonomously control its behaviour, and improve its performance over time. Furthermore, these competencies should be combined, each providing important information for other parts of the system. In this paper we present *George*, the robot depicted in Figure 1, that was built on these premises. George autonomously and continuously acquires new categorical knowledge about objects in its environment through mixed initiative dialogue with a tutor.

Continuous, interactive learning is important from several perspectives. A system operating in a real life environment is regularly exposed to new observations (scenes, objects, actions etc.). Therefore, it has to be able to update its knowledge continually based on new visual information, and information provided by a human teacher. A natural dia-

*Corresponding author. Email: danijel.skocaj@fri.uni-lj.si

logue is an appropriate way of conveying conceptual knowledge from a teacher to a robot. Assuming that the information provided by the teacher is correct, such interactive learning can significantly facilitate the learning process. Since learning is prone to errors due to unreliable robot perception capabilities, having a human in the loop can significantly improve the robustness of the learning process. By assessing the system's knowledge, the human can adapt their way of teaching and drive the learning process more efficiently. Similarly, the robot can take the initiative, and ask the human for the information that would increase its knowledge most, which should in turn lead to more efficient learning. To decide when to take the initiative and what information to ask for, the robot has to be able to reliably estimate what information is needed most, therefore to detect gaps in its knowledge, and to plan how to obtain the missing information.

Figure 1. Interactive learning scenario.

In this paper we describe how our robot George learns and refines conceptual models of visual objects and their properties, either by attending to information provided by a human tutor (e.g., H:[1] 'This is a Coke can.') or by taking initiative itself[2]. In the latter case, the robot learns by *extrospection*, i.e., by analysing the scene and using the acquired information to update its knowledge, either automatically, or after asking the tutor for additional information when necessary, e.g., R: 'Is the elongated object yellow?'. George can also initiate learning by *introspection*, i.e., by analysing its internal models of visual concepts and asking questions that are not related to the current scene, e.g., R: 'Can you show me something red?'. In both cases, the robot is able to detect the *knowledge gaps* and to plan and execute actions that provide the information needed to fill these gaps. We unify these capabilities in a highly integrated, embodied, situated robot system which includes attention-driven visual processing, incremental visual learning, selection of learning goals, continual planning, and a dialogue subsystem. In this paper we focus on the representations and mechanisms that enable interactive learning and coherent behaviour in this distributed, asynchronous and heterogeneous system.

Interactive continuous learning using information obtained from vision and language is a desirable property of many cognitive systems, therefore several systems have been developed that address this issue. Different systems focus on different aspects of this

[1]In dialogues we indicate the human with "H:" and the robot with "R:".
[2]Videos of the system can been seen at http://cogx.eu/results/george.

problem, such as the *system architecture and integration* (Bauckhage et al. (2001); Billard and Hayes (1999); Bolder et al. (2008); Briggs and Scheutz (2012); Hawes, Wyatt, Sridharan, Kopicki, et al. (2010); Karaoguz, Rodemann, Wrede, and Goerick (2012); Kirstein et al. (2009); Lallee et al. (2012); Lutkebohle et al. (2009); Mason and Lopes (2011); Sun (2007)); *learning and symbol grounding* (Belpaeme and Morse (2012); Billard and Hayes (1999); Briggs and Scheutz (2012); Chernova and Veloso (2009); de Greeff, Delaunay, and Belpaeme (2009); Deits et al. (2013); Kirstein et al. (2009); Perera and Allen (2013); Roy and Pentland (2002); Salvi, Montesano, Bernardino, and Santos-Victor (2012); Schiebener, Morimoto, Asfour, and Ude (2013); Steels and Kaplan (2000); Tellex, Thaker, Joseph, and Roy (2014)); *motivation* (Lutkebohle et al. (2009); Malfaz, Castro-Gonzalez, Barber, and Salichs (2011); Merrick (2012); Oudeyer, Kaplan, and Hafner (2007); Sequeira, Melo, and Paiva (2014); Thomaz and Breazeal (2008a)); and *social interaction* (Belpaeme and Morse (2012); Cakmak, DePalma, Arriaga, and Thomaz (2010); Fong, Nourbakhsh, and Dautenhahn (2003); Lutkebohle et al. (2009); Otero, Saunders, Dautenhahn, and Nehaniv (2008); Thomaz and Breazeal (2008a, 2008b)).

As is inevitable when building a large, heterogeneous system, different parts of our approach are related to much of this prior work. However our system solves many individual problems differently, and, crucially, *integrates the individual solutions in a novel and general way.* The main contributions of the presented work are the theory and the implementation of the *detection of knowledge gaps* in a principled statistical framework and the *curiosity driven goal formation* and *epistemic planning and execution across multiple modalities.* More specifically, our work focuses on the integration of visual perception and processing of linguistic information by forming beliefs about the state of the world; these beliefs are then used in the learning process for updating the current representations. The system behaviour is driven by a motivation framework which facilitates different kinds of learning in a dialogue with a human teacher, including self-motivated learning triggered by autonomous knowledge gap detection. Also, George is based on a distributed asynchronous architecture, which facilitates inclusion of other components that could bring additional functionalities into the system in a coherent and systematic way (such as navigation and manipulation). George is one system in a family of integrated systems that aim to understand where their own knowledge is incomplete and take action to extend their knowledge subsequently. Our objective is to demonstrate that a cognitive system can *efficiently acquire conceptual models in an interactive learning process* that is not overly taxing with respect to tutor supervision and is performed in an intuitive way.

Holistically, it is possible to compare our system to a number of other integrated systems. Previously, many of the authors of this work contributed to the PlayMate system (Hawes et al. (2007); Hawes, Wyatt, Sridharan, Kopicki, et al. (2010)). This was a robot which had many of the properties we desire for George, and served as a prototype for some of the capabilities presented below. However the PlayMate lacked a vast number of George's capabilities (including 3D vision, a structured drive system, a number of George's learning approaches, abductive reference resolution and MLN-based cross-modal binding), and its architecture, whilst comparable to George, was never robustly realised.

The University of Bielefeld's Curious Robot (Lutkebohle et al. (2009)) was developed for a very similar interactive learning scenario, and was implemented using a comparable memory-based architecture. Their design is guided by similar principles to ours, particular the idea that the robot should structure the interaction to direct learning, using both dialogue and pointing. To enable this, their system also uses a motivation system to generate learning-directed tasks. In comparison to this work George employs

much richer 3D models of objects and their possible properties, and more detailed models of corresponding beliefs. Key differences include George's ability to separate different classes of beliefs (e.g. what it has derived from sensors vs what is assumed from human utterances) and to use the absence, and quantified presence, of belief properties to plan information-gathering utterances in detail. George also generates behaviour using a general-purpose planning and execution architecture which allows it to be easily extended with appropriate actions as required (gaze control, pointing, etc.).

In addition to having a similar name, the Curious George robot from the University of British Columbia (Meger et al. (2008)) employs comparable 3D attention models to allow it to identify regions of its environment which are salient with respect to the presence of objects to be learnt or identified. However this is where the similarities end as this system does not learn objects through interaction with a human.

Recent work by Stephanie Tellex and colleagues (Deits et al. (2013); Tellex, Thaker, Deits, Kollar, and Roy (2012)) has developed the notion of generalised grounding graphs which connect natural language to the percepts of a robot using probabilistic graphical models. The use of these graphs, combined with information theory, allows their system to perform similar interactive behaviour to George: resolving references to perceptual entities across multiple utterances, clarifying ambiguous references, and, perhaps most uniquely, using the quantified results of uncertain perception (e.g. the type of object - "coke can", "bottle" - or its properties - "red", "elongated") to select the most appropriate question to ask (e.g. a polar or open question). The Tellex work employs a more detailed (mathematical) model of the interplay between interaction, perception and beliefs, but does not make the important distinction between the nature of beliefs derived from different sources. Their system also lacks multi-layered visual processing of George, or its general motivation and behaviour generation architecture. Theoretically, it would be possible to use Tellex's work in place of George's situated dialogue processing system, embedding their richer probabilistic model within our broader integrated system.

The rest of this article is organised as follows. In §2 we present the competencies and representations that allow integrated continuous learning. In §3 we describe the system that we have developed and focus on mechanisms that implement different behaviours leading to a coherent compound learning behaviour. In §4 we demonstrate the system functionality and present the results of the system evaluation. We conclude the paper with a discussion and some concluding remarks in §5.

2. 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 2 provides an overview of the main competencies in our 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.

We thus present our work by first describing the contributions made by individual components to the overall system before presenting how they interact to generate system

behaviour. Whilst this allows us to introduce techniques before they are used, it can lead to the techniques becoming disconnected from their role in the wider system. Therefore readers who prefer explanations based on even more examples may wish to instead start with either §3 or §4.1, referring back to §2 for more detailed explanations of individual techniques.

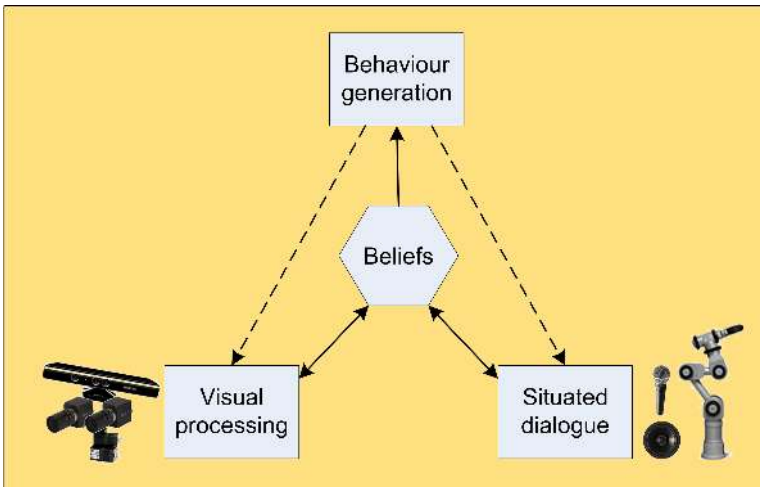

Figure 2. System competencies and relationships between them.

2.1. *Illustrative example*

Consider a scene similar to the one presented in Figure 1. The human teacher and the robot system are engaged in a dialogue aiming to teach the robot 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 teacher puts the objects in the scene and describes them or asks the robot questions about them. In our scenario, all the perceptual entities are restricted to be a single shape, (predominant) colour and type. The robot looks around, detects the objects, and processes the visual and linguistic information to understand what is in the scene and to plan how to learn about the objects and their properties.

For instance, let us suppose that the current view of the robot is depicted in Figure 3. The teacher 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: 'What colour is the tea box?'). The robot will answer the question (R: 'It is red.'), but it could also take the initiative and ask the tutor a question that would require an answer, which 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, e.g., R: 'What shape is this object?'. In such a mixed initiative dialogue the robot tries to get as much information from the human 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. An example of a scene George can learn from.

2.2. *Attention driven visual processing*

To autonomously learn visual object concepts the system needs to identify when new objects are presented to it and which parts of the scene are interested. Since the robot cannot have models for unseen objects it cannot rely on model-based detection and recognition, but requires a more general mechanism. To this end George uses a *generic, bottom-up 3D attention mechanism* suited for indoor environments.

To make the problem of generic segmentation of unknown objects tractable we introduce the assumption that objects are always presented on a supporting surface such as the table in Figure 3. Given 3D point clouds (obtained with an RGB-D sensor), our system detects (possibly multiple) supporting planes using a variant of particle swarm optimization (Zhou, Richtsfeld, Zillich, and Vincze (2011); Zhou, Varadarajan, Zillich, and Vincze (2011)). Any parts sticking out from the detected supporting plane are labeled as 3D *spaces of interest* (SOIs), i.e. something that is potentially interesting to the robot (5 individual SOIs in the case of Figure 3). SOIs are tracked over time (eliminating transient features or noise) using colour histogram, size and position information. We refer SOIs which are successfully tracked as *stable*.

As segmentation based on the RGB-D data alone can be imperfect and can include points with erroneously assigned background colour (due to shadowing effects at object boundaries), stable SOIs are augmented with a precise segmentation mask using graph cut (Boykov, Veksler, and Zabih (2001)). 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 foregorund pixels, different shape features etc.). Figure 4 shows the results of processing the scene depicted in Figure 3: segmented point cloud, detected objects and the close-up view after foveating on an object. The individual isolated objects are then subject to further processing as described in the following subsections.

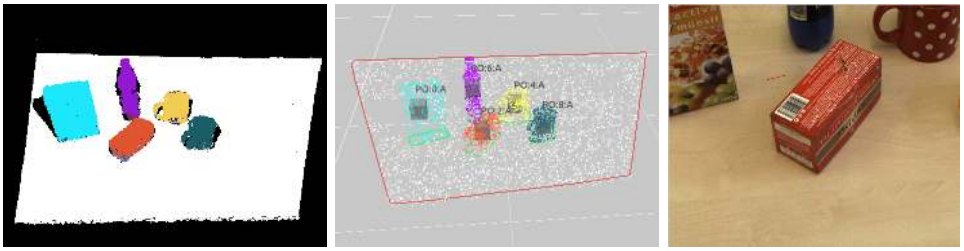

Figure 4. Segmented point cloud, detected objects, and a close-up view of a foveated object.

2.3. *Learning and recognition of object properties*

To efficiently store and generalise visual information, the visual features of object properties (such as colours and basic shapes) are represented as generative models. These generative models take the form of probability density functions (PDFs) over the feature space, and are constructed in an online fashion from new observations. This continuous learning process extracts visual data in the form of multidimensional features (e.g., multiple 1D features relating to shape, texture, colour and intensity of the observed object), and the *online discriminative Kernel Density Estimator* (odKDE; Kristan and Leonardis (2013)) that we have developed is used to estimate the PDF in this feature space. The odKDE estimates the PDFs by a mixture of Gaussians; is able to adapt using only a single data-point at a time; does not assume specific requirements on the target distribution; and automatically adjusts its complexity by compressing the models using discriminative criterion. A particularly important feature of the odKDE is that it allows adaptation from the positive examples (learning) as well as negative examples (unlearning; Kristan, Skočaj, and Leonardis (2010)).

Therefore, during online operation, a multivariate generative model (e.g. over HSL colour feature space) is continually maintained for each of the visual concepts (e.g. every colour) indicated by the tutor, and, 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 recognition of individual object properties. However, since the system is operating online, it could at any moment encounter a concept that has not been observed before (e.g., a new colour). We model the probability of this occurring with an "unknown model", which should account for poor classification when none of the learnt models strongly supports the current observation. Having built such a knowledge model and Bayesian classifier, recognition is done by inspecting the posteriori over individual concepts and the unknown model.

Such a knowledge model is also appropriate for detecting gaps and uncertainty in knowledge (Kristan, Skočaj, and Leonardis (2010); Wyatt et al. (2010)). By analysing the a posteriori probability, the system determines 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 perceived 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. §2.6 explains how this information is stored in the system beliefs to be used by higher-level cognitive processes, such as planning.

2.4. *Learning and recognition of object types*

Besides generic object properties, George also learns types of objects using the method we described in (Zillich, Prankl, Mörwald, and Vincze (2011)). We use view-based 3D object models consisting of a series of registered object views, each containing a set of SIFT (Lowe (2004)) and SHOT (Salti, Tombari, and Di Stefano (2014)) features together with their 3D position on the object surface. These views are incrementally acquired from RGB-D images, and are aligned using sparse bundle adjustment. Using texture-based (SIFT) as well as shape-based (SHOT) features allows us to represent objects characterised predominantly by their visual appearance (e.g. coke cans) or a more general

shape (e.g. mugs). Object recognition then uses RANSAC to find a matching view from the SIFT features extracted from a given RGB image, and the SHOT features extracted from the point cloud. Note that here we essentially use an object instance based recogniser as a classifier, trained from very few exemplars, which is a very hard problem in itself. We found, though, that this classifier worked well in handling the variability encountered in our scenarios, and found it a good compromise regarding number of required training examples vs. classification performance compared to more elaborated approaches like (Wohlkinger, Buchaca, Rusu, and Vincze (2012)).

Similar to the above object properties, the robot maintains not only the models themselves, but also measures of their completeness. To this end we define probabilistic measures for *observed detection success*, *predicted detection success* and *model completeness*, allowing George to quantify its current knowledge about an object and the predicted increase in knowledge for a given action (e.g. adding a new view after a change of view point). *Observed detection success* $p(o \mid c)$ is the probability of having successfully detected the object o given the detector's confidence value c. Although confidence values are often expressed in the range $0 \ldots 1$ they do not constitute actual probabilities. Therefore we learn the meaning of that confidence value in terms of a probabilistic observation function. To do this we use a series of virtually rendered views of the model acquired so far to generate training examples. *Predicted detection success* $p(o_j \mid \theta)$ is defined as the probability of successfully detecting object view o_j given an out of plane rotation θ, and is again learned using virtual training examples. To arrive at a measure of *model completeness* we take the expected detection probability over all learned views $\hat{p}(o) = \sum_{\theta,j} p(o_j \mid \theta)p(\theta)$ where prior $p(\theta)$ can take into account that certain views are less likely than others and thus possibly not even learned (such as the underside of an object). We then define the knowledge gain g when learning a new view $n+1$ as the expected increase in $\hat{p}(o)$ after learning the new $(n+1)$-th view: $g = \hat{p}_{n+1}(o) - \hat{p}_n(o)$. I.e. we tentatively add the (empty) future view to our model together with its *predicted detection success* and calculate the increase in detection probability.

We therefore represent the completeness of an object model as the expected detection probability over all object views learned so far, where the detection probability of individual views are learned from virtual training examples. As the model aligns its learned views it is possible to infer which parts of the view sphere are currently not covered. These views are knowledge gaps which the system can fill, where the knowledge gain measure introduced above quantifies the gain in closing that gap.

2.5. *Situated dialogue*

In addition to vision, the other external source of information for George is dialogue with a human tutor. In task-oriented interactions between a human and a robot, there is more to dialogue than just understanding words. The robot needs to understand *what* is being talked about and *why* it was told something. In other words, what the human *intends* the robot to do with the information in the larger context of their joint activity. To do so, we employ *continual abduction* (Janíček (2012)) to generate and verify hypotheses about the human tutor's behaviour in terms of communicative intentions, explicitly representing these in the system.

Abduction is a method of explanatory logical reasoning (Fann (1970)). 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.

In practical applications there may be many possible explanations for a fact, thus a mechanism is required for selecting the best one. This may be done through syntax (e.g. lengths of proofs), or semantics. Semantic selection can be done by assigning *weights* to abductive proofs and selecting either the least or most costly proof (Stickel (1991)), or by assigning probabilities to proofs (Poole (1993)), with the most probable proof assumed to be the best explanation. Our approach combines both weighting and probabilistic elements.

An *intention* is a goal-oriented cognitive state usually modelled as an explicit commitment to acting to achieve a goal or desire (Bratman (1987); Cohen and Levesque (1990)). George's communication system explicitly models *communicative intentions*, i.e. intentions that are related to communication[3], and uses them as a pragmatic representation of the human-robot interaction, abstracting away from the actual surface form.

Abductive reasoning over intentions in a situated context is a bi-directional process (Stone and Thomason (2003)) used in our system 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).

We employ abduction in a continual manner, explicitly modelling the knowledge gaps that arise due to uncertainty and partial observability. Our approach is based on the generation of partial hypotheses to explain the observed behaviour of other agents, under the assumption that the observed behaviour is intentional. These partial hypotheses are defeasible (i.e. open to revision) and conditioned on the validity (and eventual verification) of their assumptions. The abductive reasoning system represents knowledge gaps as partial abductive proofs. To turn partial proofs into "full" proofs, the knowledge gaps in them need to be verified or falsified.

We extend logic-programming-based weighted abduction (Hobbs, Stickel, Appelt, and Martin (1993); Stickel (1991)) by augmenting the proof procedure with the notion of *assertion* based on the work in continual planning (Brenner and Nebel (2009)). This allows the system to reason about information not present in the knowledge base, thereby addressing the need for reasoning under the open-world assumption. In planning, assertions support reasoning about information that is not known at the time of planning, encoding the expectation that the denoted information will be resolved eventually. In our logic programming approach we use asserted facts with unbound variables to both indicate that a given assertion will become a proven fact and to under-specify its arguments.

2.6. *Modeling beliefs*

By processing visual information and communicating with the human, our system forms *beliefs* about the world (Vrečko, Janíček, Leonardis, and Skočaj (2012)). 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 an amodal representation. In general a belief can be regarded as a high-level representation of an element of the physical reality, potentially grounded in one or more sensory inputs, and attributed to specific agents or groups of them. In our work, a single belief contains information about one entity, but there can be many beliefs

[3]As opposed to, for instance, the robot's purely *internal* intentions that have nothing to do with communication. See also §2.8.

about a single entity. In George, the contents of beliefs are expressed as multivariate probability distributions over feature-value pairs.

In order to support interactive learning, it is essential for beliefs to model the multi-agent aspects of knowledge acquisition and communication: a belief can be private to the robot, attributed to an external agent (e.g. the tutor), or common ground among the robot and one or more other agents. In this sense we distinguish five distinct belief categories:

Private beliefs reflect the robot's perceptions of the environment from sensory input. They are expressed in modal symbols.

Assumed beliefs are used to establish cross-agent or cross-modal common ground. They are created from private beliefs by translating modal symbols to amodal ones. In the cross-agent case the robot uses assumed beliefs to establish a common ground with another agent to facilitate communication. Thus assumed beliefs reflect the robot's assumptions about the meaning of its perceived information for another agent. In the cross-modal case the assumed beliefs establish a common ground between modalities. In both cases assumed beliefs facilitate cross-belief information fusion in later processing.

Attributed beliefs contain information that the robot attributes to an another agent (e.g. the tutor). Attributed beliefs are the direct consequence of some kind of communication with another agent. The robot is in principle able to analyze and understand the information in such beliefs, but does not necessarily agree with it (e.g. if it doesn't match the robot's own perception of the same reality).

Verified beliefs are created from attributed beliefs. They contain the acknowledged information from the attributed beliefs. Acknowledgment (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. When analyzing information from another agent, after a successful reference resolution the referring information is stored in verified beliefs (ready to facilitate possible future communication about the same subject), while the asserted information is stored in attributed beliefs (from where it can be e.g. directly used for learning).

Merged beliefs combine information from verified and assumed beliefs and represent the final amodal situated knowledge, ready for higher level cognitive processes (e.g. motivation, planning). They collect all available information, trying to maximize its reliability. Information can be merged in different ways. E.g. 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 information fusion involves merging probability distributions over feature values.

George creates private beliefs using the information from the visual subsystem (and other modal subsystems if provided). The attributed and verified beliefs are created as results of successful resolution of another agent's reference (as described in §2.7). Changes in perception are propagated in real-time through the belief structure from the private beliefs to the merged ones. In a similar manner the updates from situated dialogue processing are reflected in changes in attributed and verified beliefs (e.g acknowledgments for the attributed information). This means that the process of belief merging is repeated each time new information is propagated to the assumed belief or new attributed information is verified.

We will illustrate the belief maintenance process on an example depicted in Fig. 3. Let us focus on the blue elongated object and let us assume that the robot was able to recognize the colour and shape of it, but it failed to recognize its type ("a bottle"). Having observed the scene and after hearing the tutor saying: 'The blue object is a bot-

tle', the following beliefs are formed. First, a private belief is formed, containing the modal representations (directly related to the corresponding visual features) for "blue" and "elongated". These symbols are translated to amodal symbols of blue and elongated (using a previously learned mapping) that are stored in an assumed belief. After processing the tutors statement first an attributed belief is created containing information about the "blue" and "bottle", where "blue" forms the referring part of information, while "bottle" represents the assertive part of information. After the referring part of information is used for reference resolution (as it is described in §2.7), a verified belief is created containing this information ("blue"). Finally, a merged belief is created by merging the information from the assumed and verified beliefs, taking the amodal symbol for elongated shape from the assumed belief and the amodal symbol for blue colour from both assumed and verified beliefs.

2.7. *Binding and reference resolution*

In order to learn from situated dialogue, it is essential that an interactive robot is able to perform *reference resolution*. In our case George uses reference resolution to relate information provided by a human tutor to its own perceptions. For this task we use a *cross-modal binding* mechanism. Cross-modal binding is the process of combining two or more modal representations of the same physical entity into a single multi-modal representation. In (Vrečko, Leonardis, and Skočaj (2012)) we presented a cross-modal binding and learning system formulated in *Markov logic networks (MLNs)*. MLNs combine first-order logic and probabilistic graphical models in a single representation (Domingos (2007); Richardson and Domingos (2006)). In our approach to binding, the MLN knowledge base consists of a set of first-order logic formulae (rules) with weights attached. This knowledge base encodes cross-modal knowledge, providing the base for the binding inference.

George utilises our MLN binding approach within its belief layer. The MLN knowledge base represents general relations between modal object concepts (e.g. colour, shape etc.). Private and assumed beliefs are used to instantiate the rules from this knowledge base to the *Markov network* graphical model. The graphical model therefore encodes cross-modal relations between the specific object instances that are currently perceived by the robot (Vrečko, Leonardis, and Skočaj (2012)). The resulting MLN model can continuously adapt to the changes in the beliefs through online inference (MCMC sampling) as it processes information from two distinct modalities via the belief layer: information about perceived entities, and information from the dialogue subsystem. When the dialogue subsystem recognises a referring expression in the tutor's utterance, it forwards the referring information to the MLN engine. The inference result, a probability distribution over perceived entities, is used by the dialogue subsystem to help determine the interpretation of the tutor's utterance, then the corresponding beliefs are updated as described in §2.6.

2.8. *Motivation and planning*

George must perform multiple, possibly interleaved, goal-directed activities. As a system that must fill gaps in its own knowledge, it is important that is able to generate and manage its own goals, as the opportunities available to it at runtime may be unknown or unpredictable at design-time. To address this we have utilised on our previous design for a *motivation framework* (Hawes (2011); Wyatt et al. (2010)). This framework encodes

the *drives* of the system (the general types of things it wants to achieve) as a collection of *goal generators*, each of which generates particular types of goals for the system based on the results of situated dialogue plus the current belief state. Each individual goal is a (partial) description of a desired future state for the robot (e.g. one in which it knows the colour of a newly visible object). Before these goals can be *activated*, i.e. made the target of planning and execution, they must pass through a management system that selects which of the many possible goals should be pursued by the system. This management step is necessary to allow the robot to prioritise important, or more appropriate goals (given its drives, context etc).

The goal generators in George create the goals necessary to engage in situated dialogue with a human tutor and to learn about its surroundings. A goal generator monitors the communicative intentions created by the dialogue system as it interprets the utterances made by the tutor (see §2.5). Depending on intention content, this generator creates goals to answer polar or open questions about objects, or to perform tutor-driven learning. Each of these goals contains a reference to the single merged belief for the object referenced by the intention, plus additional intention-specific information. An additional 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 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. Further goal generators inspect the beliefs created from entries in the visual subsystem, including proto-objects and visual objects (as described in §3.2.1); and concept models. These create goals to generate visual objects, learn features and improve the models respectively.

The activation of goals in our system is based on a priority hierarchy of drives. Each level in the hierarchy controls a type of behaviour we have identified that George should perform. The highest priority drive is to *respond to the human*. This is followed by the drive to fill gaps in knowledge via *extrospection* (i.e. inspecting the world external to the agent). At the lowest level is the drive to fill knowledge gaps by *introspection*. Goals of a particular priority suppress the activation of all goals with lower priorities and are suppressed by all goals with higher priorities. This is accomplished using a simple *attention filter* in our framework (Hawes (2011); Wyatt et al. (2010)). Goals that pass through this filter enter into the management system. Here they can be ranked according to heuristics provided by their goal generators and the top ranked goals are activated.

The motivation framework provides George with a means to trigger and focus learning. As such it can be compared to artificial curiosity systems, such as those pioneered by Schmidhuber (Schmidhuber (1991)) and Oudeyer (Baranes and Oudeyer (2009); Oudeyer et al. (2007)). The key difference in our approach is that curiosity drives are handled as part of a larger deliberative system that can flexibly switch between task-driven and curiosity-driven goals. The drawback of our approach is finding metrics for evaluating the relative benefits of different goals, an issue which is more neatly formulated in work dedicated solely to artificial curiosity, such as that of Schmidhuber. We use a drive hierarchy partly to address this problem as it allows George to distinguish between goals which are not directly comparable numerically. It would, however, be perfectly possible to substitute such a universal curiosity measure for our set of specific curiosity measures for each problem.

Planning is performed for activated goals on a problem description generated from the system's belief state. Plan execution, execution monitoring and replanning is managed via a collection of action interfaces which trigger individual components in the modality-specific subsystems. We use the Fast Downward (Helmert (2006)) planner, a state of the

art planning system based on heuristic forward search. We extended it 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 (Brenner and Nebel (2009)).

Dialogue with a tutor, in the form of asking or answering questions, plays an essential role for all three of George's drives. The planner must therefore find plans that establish common ground with the tutor about the object they discuss. Initial ambiguity in a dialogue is represented by having multiple objects as the referent of a linguistic statement, alongside a goal to only have a single referent. The planner can then predict the effects of *clarification actions* on its knowledge about the reference. It uses these actions to create a plan which it expects to remove the ambiguity and leave only a single referent which will then be the target of a dialogue.

George has two types of actions for clarifying a reference: describing the object verbally, or pointing to it with its arm. As we regard a verbal description as the cheaper one, George will always try to describe the object in question if it has some property that is unique among all objects and where human and robot have already established common ground, and it will choose to use the arm otherwise. Once George and its tutor know which object their discussion is about, the planner determines the correct answer or question from the belief state, and triggers learning if necessary. Many examples of the behaviour generated by George's planning system are included in the results in §4.

3. Integrated system and behaviour mechanisms

3.1. *Integrated system*

We integrated the competencies described above in a robotic system. The design and implementation of our integrated system is based on CAST, the CoSy Architecture Schema Toolkit (Hawes and Wyatt (2010)). The schema is a distributed working-memory model composed of several subarchitectures, each implementing a different functionality. A subarchitecture (SA) contains one or more components each running in its own thread. The components communicate through the working memory (WM). When the state of a component changes it either adds an object of a known type containing the relevant information to the WM, updates an object in the WM or deletes an object from the WM. Another component can register with the WM to receive an event whenever a change to an object of a certain type occurs. This allows links between multiple components to be established and for information to be passed accordingly. This working-memory approach shares features with general-purpose cognitive architectures (e.g. Laird, Newell, and Rosenbloom (1987)), robot architectures such as 3T Bonasso et al. (1997), and older blackboard-based systems (e.g. Erman, Hayes-Roth, Lesser, and Reddy (1980)). Our architectural approach is described in more detail in existing work (Hawes and Wyatt (2010); Hawes, Wyatt, Sridharan, Jacobsson, et al. (2010)). As depicted in Figure 5, George is composed of six CAST SAs. This figure presents George from the system point of view and reveals the complexity of the system and relations between the individual components that are described in this section.

The *Visual SA* processes the scene as a whole using a Kinect RGB-D sensor and a narrow field-of-view Point Grey Flea 2 camera. The Visual SA identifies SOIs where potential objects are detected and processed as described in §2.2. Information extracted from SOIs is then used for learning as required, as described in §2.3 and §2.4. The attention-driven visual processing also makes use of the Direct Perception pan/tilt unit

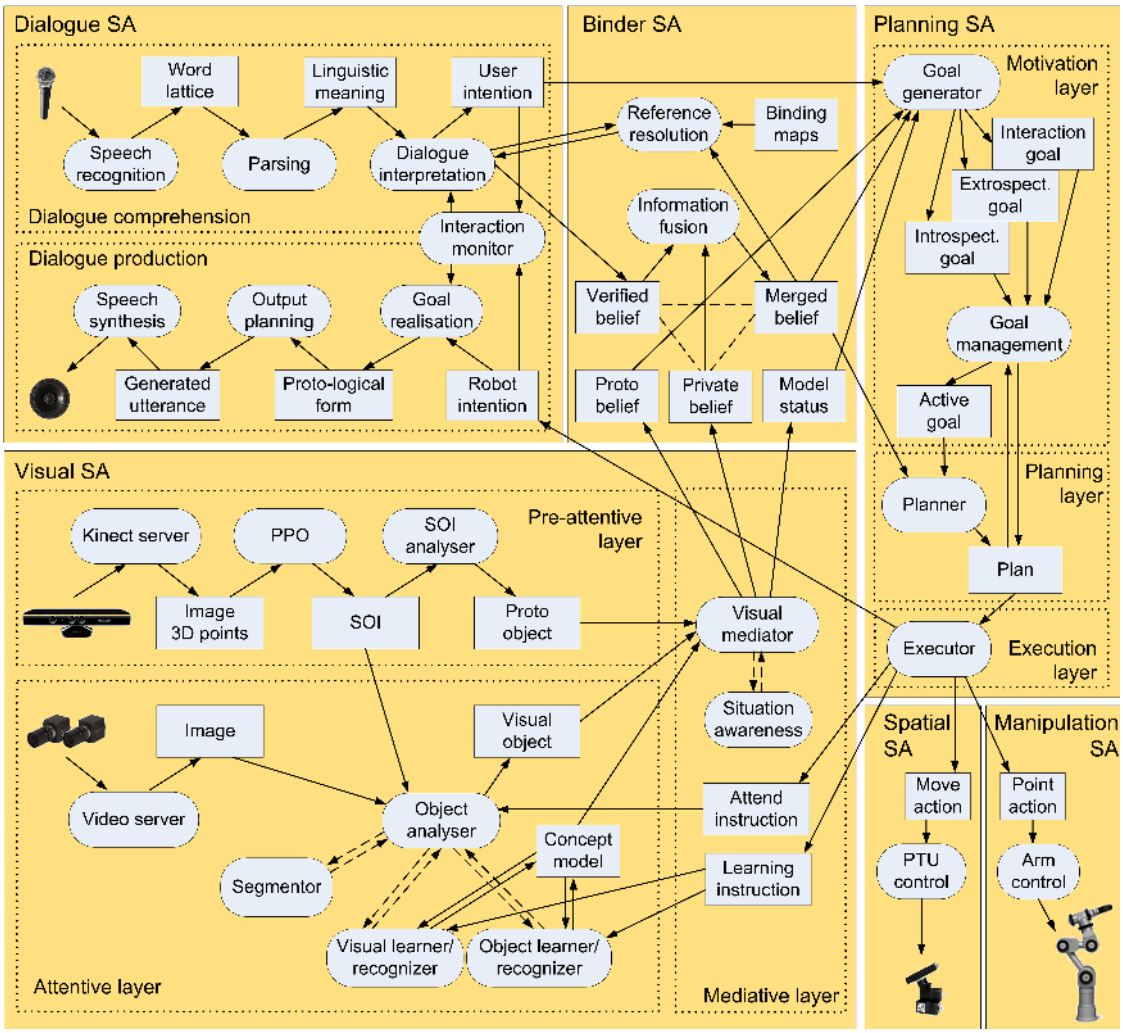

Figure 5. Schematic system architecture. Components are rounded boxes, exchanged data structures are rectangles, and arrows indicate information flow.

from the *Spatial SA* for bringing SOIs into the centre of attention.

The *Dialogue SA* provides situated dialogue processing capabilities. The system uses third party software for speech recognition, the Mary TTS system for production [4], and the techniques presented in §2.5 and §2.7 for the recognition of the tutor's intentions and realisation of the robot's intentions in the situated context. The robot also uses the Neuronics Katana 6M 5DOF robot arm (from the *Manipulation SA*, controlled via Golem (Kopicki (2010)), for pointing at the object in the scene to establish a common ground with the tutor.

All of the beliefs are collected in the *Binder SA*, which represents a central hub for gathering information from different modalities (subarchitectures) about the currently perceived objects (as described in §2.6). The beliefs are monitored by the *Planning SA*, which generates the robot behaviour as described in §2.8. The appearance of beliefs trigger goal generators to produce the learning goals. The complete set of beliefs is used to provide the planning state. Finally, during execution, action requests are sent to the Visual, Spatial, Manipulation, and Dialogue SAs to perform actions that generate the

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

desired behaviour. The mechanisms that drive these behaviours are described in the following subsection.

3.2. *Basic behaviours*

The George system is complex, heterogeneous, and integrated. This means that even its basic behaviours require functionalities distributed across several subarchitectures. The behaviours also require that different functionalities are executed in parallel, but are still kept synchronised. In the following we will briefly describe the mechanisms that implement these different behaviours. These mechanisms are depicted in Figure 6. Here, only the main processing flows are sketched, and only the major components or data structures are emphasised (encircled); for details refer at Figure 5. Every behaviour is triggered by a particular event in a component or data structure; they are marked with a thicker circle.

3.2.1. Mechanisms for visual perception

There are two main behaviours that provide the robot with the visual information. The first one is bottom-up and is triggered by changes in the scene, assuring that the objects that are brought in the view of the robot are analysed as well as possible. The second one is top-down and is triggered by the motivation subsystem, assuring that the robot looks around and analyses the entire scene. Both mechanisms are governed by the *extrospection drive* as they relate to understanding the external world.

3.2.1.1. ***Attention mechanism.*** As outlined in §2.2, George utilises bottom-up visual processing based on plane pop-out attention. This mechanism, which leads to the generation and tracking of SOIs, is always running. As mentioned previously, SOIs are tracked to maintain stable percepts during changes of lighting or small object or camera movements. However, SOIs are not maintained when George looks away from the scene and back again, as they are intended to provide the trigger for further processing, rather than endure as percepts in their own right.

Whenever a SOI is found and tracked, a *proto-object (PO)* is generated (as depicted in Figure 6(a)). A view cone is associated with the PO, indicating a potential close-up view of the PO. When the Planning SA detects a belief about a PO it generates a goal to attend to it for further analysis. If this goal is activated, execution of the subsequent plan results in a movement of the pan-tilt unit to foveate the PO, bringing it into the centre of the higher resolution camera, where it is analysed, i.e. its precise outline will be segmented as an image region of interest (ROI) and object properties (colour, shape) extracted. Furthermore learned object recognisers are run on the ROI in the high-resolution image, resulting in a label if the object is already known. The generated properties and label are stored in a *visual object* (VO), represented as a *private belief* in the Binder SA. When a newly detected SOI matches an already existing PO (e.g. when the camera moves back to a previously analysed part of the scene) it can be re-associated with VO rather than triggering repeated processing.

3.2.1.2. ***Exploring the scene.*** George has a limited view of the world. There may be objects in front of it but just out of range of its visual system. To make sure George does not miss such objects it has a goal generator which motivates it to move its camera, allowing it to perceive previously unviewed parts of the scene. This generator is triggered

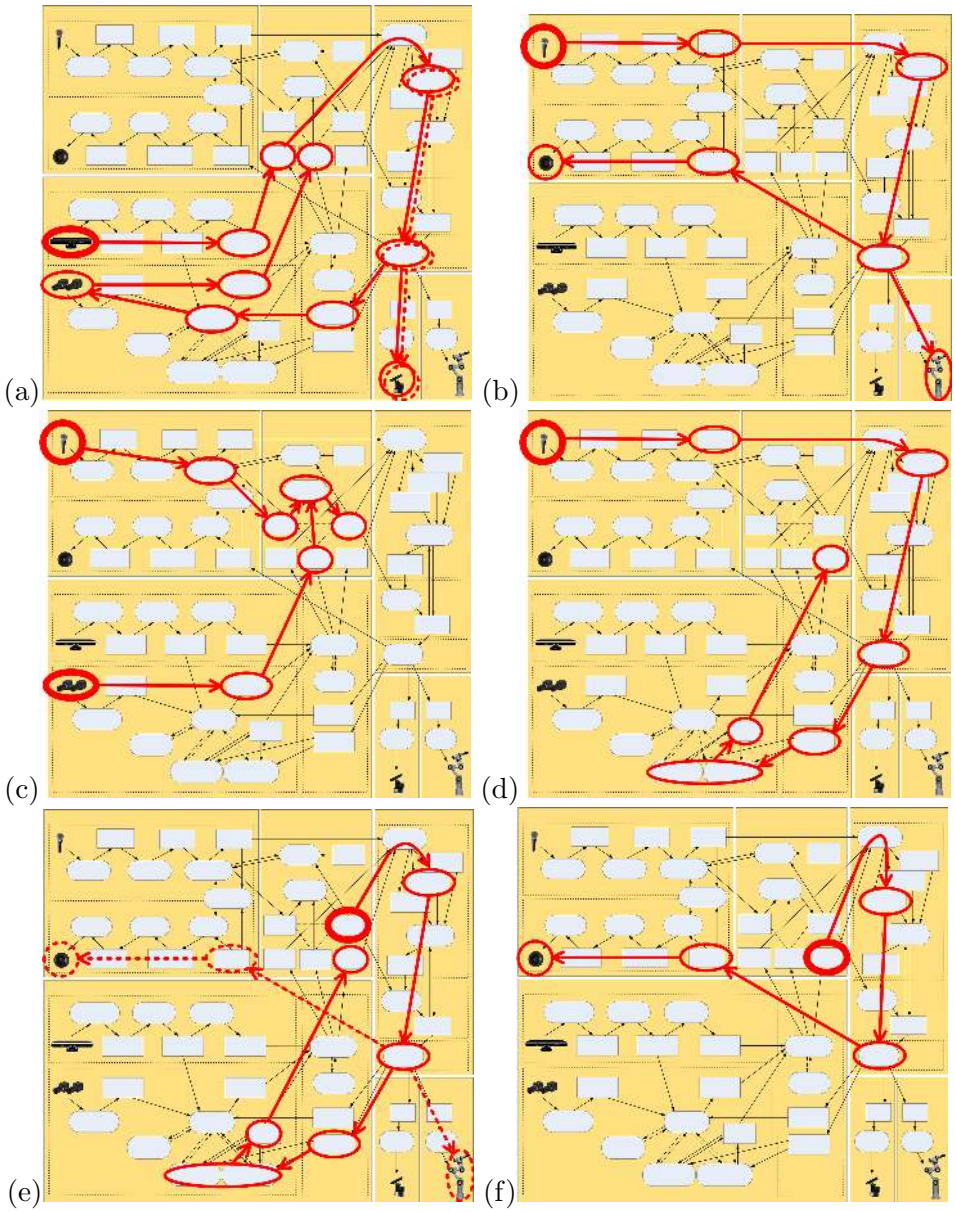

Figure 6. Behaviour mechanisms for components in Figure 5: (a) Attention and exploration mechanisms. (b) Answering tutor's requests. (c) Merging multi-modal information. (d) Situated tutor-driven learning. (e) Situated autonomous and tutor-assisted learning. (f) Non-situated tutor-assisted learning.

after a fixed window of system inactivity, and causes the generation of a plan to take a small number of random views of the scene. This mechanism is depicted with dashed lines in Figure 6(a). If a new view yields a SOI and PO then the attention mechanisms described above is triggered, potentially yielding a new VO. This process interrupts the scene exploration due to PO analysis having a higher position in George's drive hierarchy than scene exploration.

3.2.2. Tutor initiated interaction

One of the main capabilities of the system is interaction with a human tutor. Interaction can be triggered by the tutor or by the robot. We first present the mechanisms that govern the interaction initiated by the tutor, either by asking the robot to execute an

instruction or to answer a question; or by giving the robot useful information that can be used for learning. These mechanisms are triggered by the system's *interaction goals*.

3.2.2.1. ***Acting on the tutor's requests.*** The communication subsystem monitors for utterances from the tutor at all times. Whenever an utterance is detected, its surface form is analysed to provide the underlying communicative intention. As depicted in Figure 6(b), after input is detected (either by speech recognition or text input), the word sequence is parsed, assigning a semantic structure to it. This structure is then passed on to the context-sensitive intention recognition module which resolves references and connects the speech act to the previous discourse. The resulting intention then contains a reference to the object in question, and to previous intentions already present in the working memory. The resulting intention is then written to the Binder SA, triggering goal generation by motivation framework. This framework decides whether (and when) to make the goal active, reacting to the tutor's input. Due to the position of interaction at the top of George's drive hierarchy, interaction goals are acted upon immediately, except when an interaction is already underway (in which case it is queued for subsequent activation).

George can recognise both assertions about the environment (e.g. H: 'The red object is a coke can.') and questions (H: 'What colour is the coke can?', H: 'Is the coke can blue?'). The generation of George's own utterances is controlled by the planner. This holds both for answers to the tutor's questions (R: 'It is blue.') and when George takes the initiative (R: 'Could you show me something red?').

When interpreting an utterance the Dialogue SA extracts the information about the current scene it provides, and relates this to the beliefs in the Binder SA. Different beliefs are related and merged in the processes of *reference resolution* (see §2.7) and *information fusion* (see §2.6). In Figure 6(c) we can see how reference resolution relates information from the Dialogue SA (verified beliefs) to perceptual information in private beliefs. Information from both sources is then merged by the process of information fusion. The final result of these processes is a collection of *merged beliefs* that are used for further higher-level processing.

3.2.2.2. ***Situated tutor-driven learning.*** A specific case of the above is when the tutor explicitly tries to teach the robot something, which we refer to as situated tutor-driven learning. Such a learning act is initiated when (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. As depicted in Figure 6(d), 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 carry out the actions to update the internal *visual models*, resulting in an updated *model status* belief containing key information about the visual models.

3.2.3. Extrospective learning mechanisms

To make learning efficient, a cognitive system has to be able to exploit learning opportunities, not just passively wait for a tutor's learning instructions. It should actively look for, ask for, and use the information that would help to extend its knowledge. George aims to extend its knowledge about visual concepts by taking opportunities to minimise the uncertainty of its perceptions of the objects in the scene. In this case, a learning opportunity is presented by a perceived object plus some associated information (e.g. the absence or uncertain presence of a particular feature). Learning mechanisms associated with such opportunities are triggered by George's *extrospective drives*. This is realised by goal generators monitoring the *merged beliefs* in the BSA; if they contain information that can be exploited for learning, goals are generated for each possible learning opportunity.

3.2.3.1. ***Situated autonomous learning.*** If a *merged belief* contains only the information provided by the Visual SA and this information is reliable (the visual concept has been recognised with high a confidence), motivation triggers an autonomous learning cycle. The representations of the corresponding visual concepts are automatically updated, resulting in an updated *model status* belief (as depicted in Figure 6(e)). In the case of a very confident recognition (with a probability very close to 1), such an update is not necessary because the current representation can describe the object perfectly well. However, in the case of slightly less reliable recognition, it makes sense to update the knowledge in this way, since it will adapt to the perceived object, increasing the confidence of recognition of the same (or similar) objects in the future. However, there is always a danger of incorporating erroneously recognised information into the representations 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 the tutor.

3.2.3.2. ***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 motivation subsystem creates a goal based on the merged belief containing the information from the private belief only. The planner drives the robot to asks about the object property with the highest *information gain* (as defined in §2.3), since it expects that the model of the corresponding object property will benefit most if it gets the information requested. In the absence of attributed information, the planner generates a more complex plan to ask questions about missing information. The execution subsystem generates a corresponding robot intention, which is further managed by the Dialogue SA, resulting in the synthesis of an utterance (the dashed branch in Figure 6(e)). Depending on the confidence in the recognition results the planner can select between polar questions (that can be answered with 'yes' or 'no') when recognition confidence is high (e. g. R: 'Is the colour of this object red?') and open questions (that require a label for the answer) when confidence is low (e. g. R: 'What is the colour of this object?'). The planner is able to identify cases when the robot cannot unambiguously refer to the particular object verbally. To solve such situations it generates a plan which drives the robot to point at the object in question to establish a common ground with the tutor. After the tutor answers, the workflow is similar to tutor-driven learning.

3.2.4. Introspective learning mechanisms

Even in the absence of situated learning opportunities, 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 most interesting). This behaviour is based exclusively on the introspection of the existing property models. From a pool of currently maintained property models the robot selects the one considered to be the least adequate (typically inadequately sampled) and initiates an action that tries to obtain new samples to improve it.

3.2.4.1. ***Non-situated tutor-assisted learning.*** In non-situated tutor-assisted learning, 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?'). Using model introspection, the robot tries to influence the quality of the potential new object. Model introspection is performed in the Visual SA and the results propagated to the belief layer in the epistemic structure *model status* which contains key information about the models maintained by the visual learner. The most important information is again the *information gain* that in this case estimates the reliability of a model in general, not related to a particular object in the scene.

3.3. *Compound behaviour*

In normal interactions, different behaviours could be triggered simultaneously, so there is a need for a mechanism that selects among them to assure coherent compound behaviour. As described in §2.8, we model this compound behaviour by assigning different priorities to the main drives that raise different goals. The motivation component opts for the goals with higher priorities. Among the goals from the same priority level the planner selects which one to pursue based on the gains it provides (how much the system is expected to benefit if the goal is fulfilled) and the cost of the plan to achieve it. The information about the gains is stored in the beliefs and is based on an analysis of the models of the visual concepts and objects that are currently present in the scene. Table 1 lists three main drives that trigger the behaviours described above.

The *interaction drive* has the highest priority, since we want that the robot reacts to tutor's assertions or requests promptly; this is a basic requirement for a natural robot-tutor dialogue. Goals to explore the scene and to learn as much as possible about the objects currently presented in the scene are part of the *extrospection drive* as they relate to understanding the external world. They will be suppressed by interaction goals but at the same time they are prioritised above model introspection. The robot first tries to learn as much as possible about the objects in the current view by attending them and updating the knowledge based on the obtained information. When these goals are not active any more, the exploration behaviour is triggered to explore the wider scene as it can yield new objects which can be learnt about.

Table 1. Priority levels.

Interaction drive	Answering tutor's requests Situated tutor-driven learning
Extrospection drive	Attention mechanism Situated autonomous learning Situated tutor-assisted learning Exploring the scene
Introspection drive	Non-situated tutor-assisted learning

Non-situated tutor-assisted learning is triggered by goals from the *introspection drive* at the lowest priority level. As such it is only carried out when no other goals are active (i.e. when all visible objects have had their properties learnt and no scene exploration is necessary). Therefore, when the robot doesn't have anything else to do, it asks the tutor to show it an object with particular visual properties that would potentially increase the robot's models of these properties most.

We chose this particular drive prioritisation to reflect the desired behaviour of the robot: it should always try to respond to the human, then try to understand the scene in front of it (as this will be the subject of future interactions), then try to understand the world in more general terms (e.g. through improving its models).

4. Experimental results

In this section we present experimental results which demonstrate the properties of the George system in general, and our approach to interactive, situated learning in particular. To illustrate system behaviour during the learning process, we first present a sample dialogue between a human tutor and the robot. Then we present quantitative results, obtained by observing the robot's behaviour in a similar scenario.

4.1. *Example dialogue*

A good way of describing the behaviour of the developed system is to present a sample dialogue between the robot and the human tutor during learning of visual concepts, such as colour, shape and object models. The robot is asked to recognize and describe the objects in a table top scene, of which there are up to five on the table at any time. The human can move or remove objects from the table during the dialogue, and teach the robot about the objects by describing them. Initially the tutor drives the learning, but after a while, the robot takes the initiative, and is able to learn either without verbal feedback, or by asking the tutor for clarification when necessary. To achieve this the robot must establish a common understanding with the human about what is in the scene, and verbalize both its knowledge and knowledge gaps. In a dialogue with the tutor, the robot keeps extending and improving the knowledge. To test what the robot has learned the tutor asks questions about the scene. The goal of learning is for the robot's representations to be rich enough to correctly describe the scene.

Consider an empty table. The tutor puts an object down and the robot looks at it by applying the *attention mechanism*.

H: Do you know what this is?

R: No.

At the beginning the robot knows nothing about any object. *Situated tutor-driven learning* is therefore suitable during these initial stages, since the robot has to be given information to reliably initialise its visual concepts.

H: This is a red object.

R: Let me see. OK.

After George gets this information, it can initiate its visual representation of redness. After several such learning steps, the acquired models become reliable enough that they can be used by George to refer to individual objects, and to understand references by the human. From this point on there can be several objects in the scene at the same time, and by applying the mechanism for *answering tutor's requests* George can understand and answer questions about them:

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 the tutor questions. It will typically do this when it is able to detect an object in the scene, but is not certain about some or all of its properties. In such *situated tutor-assisted* learning there are two types of uncertainty and gaps in knowledge. If the object does not fit any previously learned models, the robot considers there to be a gap in its knowledge and asks the tutor to provide information about its novel property:

R: What colour is this object?

H: It is yellow.

R. OK.

In the second case, the robot is able to associate the object with a particular model, however the recognition is not very reliable. Therefore, the robot asks the tutor for clarification:

R: Is this red?

H: No. This is yellow.

R. OK.

After the robot receives the answer from the tutor, it corrects (unlearns) the representation of the concept of red and updates the representation of yellow.

In a similar case as above, but if the recognition of an object is more reliable, George updates the models without asking a question, utilising the mechanism for *situated autonomous learning*. Since there is no verification from the tutor, George is very conservative when using this learning mechanism.

In this way George attends and processes all the objects it detects in the scene. It therefore knows everything it can about these objects. It then *explores the scene* by looking around for new objects. Let us assume that there is not one. Since there is no new object it could learn about, it tries to detect a gap in its knowledge by introspection. By using the mechanism for *non-situated tutor-assisted learning* it asks the tutor to show it an object to help it improve its knowledge.

R: Please, show me something blue.

H puts an object at the table. H: This is a blue object.

R: Thank you.

George can refer to objects verbally or by pointing. Let's say that the tutor empties the table and then puts two objects in the scene. George detects and attends both. If it can recognise one object attribute, which is not equal for both the objects in the scene, it can refer to that object verbally.

R: Is the Pepsi can blue?

H: Yes, it is.

R: Thank you.

However, if it cannot distinguish between the two objects using verbal description, it refers to an object by pointing.

R points at one object. R: What colour is this object?

H: It is yellow.

R: Thank you.

In the same way, George can also choose between using verbal description or pointing for clarification when the tutor asks ambiguous questions.

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

H: Do you know what this is?

R: It is a blue object.
H: What is the green object?
R: It is an ice tea bottle.
H: What shape it is?
R: It is elongated.

4.2. *System evaluation*

It is very difficult to consistently evaluate such a complex, heterogeneous and asynchronous system. The system can exhibit different behaviours 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 of different properties placed on different positions. We ran this scenario ten times with the real robot and 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.

4.2.1. Scenario setup

The scenario setup was similar to the one shown in Figure 1. The object locations were constrained to ten fixed places spread across the table. Eighteen ordinary household objects with one predominant colour were used in the experiment. They are depicted in Figure 7. We considered three concepts (colour, shape, type) and various values of these concepts. Therefore, every experiment was characterized by:

- Objects o_i: three objects selected among the objects depicted in Figure 7.
- 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, ...*$\}$.

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

The robot's knowledge was rather weak at the beginning of each run of the scenario; only weakly pre-trained models were loaded to enable the robot to be better engaged in dialogue with a tutor.

4.2.2. Actions

The experimental interaction was dictated by a fixed sequence of actions performed by the tutor. The actions available to the tutor are presented in Table 2. During the interaction, the robot was expected to reply with the actions presented in Table 3. In general several more robot actions are possible, and other types of tutor actions are supported. However, in this experiments only the actions presented in Tables 2 and 3 are considered. They are diverse enough to lead to different robot behaviours, but still sufficiently limited to enable a consistent and controlled experiment.

Figure 7. Objects used in the experiment.

Table 2. 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 3. 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?*

4.2.3. Script

The script of the experiment is presented in Table 4. The non-indented lines present the tutor's actions, while the lines with expected robot actions are indented. We repeated this script ten times. At every run different objects were positioned in different places and different concepts were discussed; we therefore varied the variables o, p, c, and v presented in §4.2.1.

In the definition of actions analyseAsk(o) and answer(o, c, v) in Table 4 '/' means 'do nothing'. The robot would select which action to undertake (what kind of question to ask or to automatically update the knowledge) based on the reliability of the current observation. The tutor would in these cases reply adequately; either by answering the robot's question or by taking the initiative and telling the robot the information about

Table 4. Scenario script.

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

the object without being asked.

According to the script the tutor places two objects on the table at the beginning of every run. They are positioned sufficiently apart, so that only one of them is in the current camera view. The robot analyzes the visible object and, based on the analysis results, updates the knowledge autonomously or after obtaining additional information from the tutor. Then the robot looks around in search for more objects. When it finds the second object, it processes this object in a similar way. After the robot observes that there are no other objects on the tabletop, it asks the tutor for a new one, possibly with the property that it is currently most interested in. After the tutor complies, the robot processes the new object in a similar fashion as the previous ones.

At the end, the tutor verifies the robot's knowledge by asking three questions about the properties of the objects on the table. The first question is unambiguous and the robot is expected to answer immediately. For an example consider the scene depicted in Figure 8; if the robot recognizes the colours and types (names) of the objects, this question might be H: 'What shape is the blue object?'. The second question is ambiguous, however it can be disambiguated by referring to some other object property or to the object name (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 cases the robot is expected to perform the adequate actions and to answer the question.

Figure 8. A sample scene from the robot's viewpoint.

4.2.4. Experimental results

The scenario covers all seven different mechanisms presented in §3.2. The evaluation was performed based on comparison of the expected and the actual behaviour of the robot. Table 5 present the results grouped according to seven mechanisms. It lists the lines in the script presented in Table 4 that implement the individual mechanisms and reports the number of times the specific actions were expected to be triggered (#exp.) and the number of times these actions were actually successfully executed (#exec.), as observed by the experimenter.

The system exhibited a good performance for all evaluated mechanisms. The attention mechanism was successful; it was triggered whenever it was expected. The objects were of adequate sizes, and they were not occluding each other, so the detection of the objects was very reliable. The system also explored the scene whenever it was necessary.

In this experiment we did not evaluate the quality of the models built and the success of the recognition actions. Our primary goal was to evaluate the complete integrated system (i.e. if the learning and recognition mechanisms executed when they were expected). The results show that all learning mechanisms were almost always triggered and executed correctly; the learning failed only once in 75 cases.

Also 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 problems were observed when pointing was required to disambiguate the reference in the tutor's question. On two occasions the execution of the pointing action failed (along with the subsequent re-tries). 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 run, 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, but in our case we took it as a failure, since the system had not exhibited the expected behaviour. The system 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).

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

Table 5. 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

5. Conclusion

In this paper we presented an integrated system for interactive continuous learning of categorical knowledge in dialogue with a tutor. We briefly showed the proposed representations and more thoroughly described the implemented mechanisms that enable such

kind of behaviour. The main contributions of the presented work are the theory and implementation of the detection of knowledge gaps in a principled way and the curiosity driven goal formation and epistemic planning and execution across multiple modalities. We presented how the beliefs about the world are created by processing visual and linguistic information and how they are used for planning the system behaviour with the aim of satisfying its internal drives – to respond to the human and to extend its knowledge by extrospection and introspection. We described the hierarchy of mechanisms that implements a coherent compound learning behaviour. We demonstrated these principles in the case of learning conceptual models of objects and their visual properties.

During our research, we have made several contributions at the level of individual components, as well as at the system level. Several components implementing individual competencies have been developed including bottom-up visual attention mechanism based on plane-pop-out object detection and tracking, odKDE-based learning of object properties, incremental view-based learning of object models, abduction-based situated language processing, MLN-based reference resolution, belief-based information fusion, advanced motive management and goal generation, and continual planning and execution. In this paper we presented how we integrated all these competencies into a coherent and efficient system capable of mixed-initiative learning. Such an integrated robotic implementation enables both the development and evaluation of the entire system, as well as the analysis and testing of the individual components, thus facilitating research on the system, sub-system, and component level.

George is based on a distributed asynchronous architecture, which facilitates inclusion of other components that would bring additional functionalities into the system in a coherent and systematic way, such as navigation and manipulation. This would increase the possibilities of interaction with the environment and enable the robot to acquire novel information in an even more active and autonomous way. Here, the detection of knowledge gaps and planning for actions that would help to fill these gaps would play an even more important role and would enable more autonomous and efficient robot behaviour. In parallel to George, we were also developing another robot demonstrator, Dora (Hanheide et al. (2011)), which implements some of these additional functionalities. This robot is motivated to fill its gaps about the extent of space or the categories of rooms. The main objectives of this robot are to explore the space (e.g., an apartment) and to search for certain objects in it in an intelligent way by using the common-sense and acquired knowledge about the space, objects, and relations between them. Although the tasks of the two robots are quite different, they share most of the underlying functionalities, demonstrating the generality of the developed solutions. Building on these functionalities and integration approach, our final goal is to produce an autonomous robot that will be able to efficiently learn and adapt to an ever-changing world by capturing and processing cross-modal information in an interaction with the environment and other cognitive agents.

Funding

The research leading to these results received funding from the European Community's Seventh Framework Programme [FP7/2007-2013], grant agreement No. 215181, CogX.

References

Baranes, A., & Oudeyer, P.-Y. (2009, October). R-IAC: robust intrinsically motivated exploration and active learning. *IEEE TAMD*, *1*(3), 155 –169.

Bauckhage, C., Fink, G., Fritsch, J., Kummmert, F., Lomker, F., Sagerer, G., & Wachsmuth, S. (2001). An integrated system for cooperative man-machine interaction. In *IEEE international symposium on computational intelligence in robotics and automation* (p. 320-325).

Belpaeme, T., & Morse, A. (2012). Word and category learning in a continuous semantic domain: comparing cross-situational and interactive learning. *Advances in Complex Systems*, *15*(03n04), 1250031.

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

Bolder, B., Brandl, H., Heracles, M., Janssen, H., Mikhailova, I., Schmudderich, J., & Goerick, C. (2008). Expectation-driven autonomous learning and interaction system. In *Humanoids 2008. 8th IEEE-RAS international conference on* (p. 553-560). Daejeon, South Korea.

Bonasso, R. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P., & Slack, M. G. (1997). Experiences with an architecture for intelligent, reactive agents. *J. Exp. Theor. Artif. Intell.*, *9*(2-3), 237-256.

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

Bratman, M. (1987). *Intentions, plans, and practical reason*. Harvard University Press, Cambridge, MA, USA.

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

Briggs, G., & Scheutz, M. (2012). Multi-modal belief updates in multi-robot human-robot dialogue interactions. *Linguistic ad cognitive approaches to dialogue agents*, 67.

Cakmak, M., DePalma, N., Arriaga, R. I., & Thomaz, A. L. (2010). Exploiting social partners in robot learning. *Autonomous Robots*, *29*(3), 309–329.

Chernova, S., & Veloso, M. (2009). Interactive policy learning through confidence-based autonomy. *JAIR*, *34*(1).

Cohen, P. R., & Levesque, H. J. (1990, March). Intention is choice with commitment. *Artificial Intelligence*, *42*, 213–261.

de Greeff, J., Delaunay, F., & Belpaeme, T. (2009, June). Human-robot interaction in concept acquisition: a computational model. In *International conference on development and learning, ICDL 2009* (p. 1-6).

Deits, R., Tellex, S., Thaker, P., Simeonov, D., Kollar, T., & Roy, N. (2013). Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction*, *2*(2), 58–79.

Domingos, P. (2007, August). Toward knowledge-rich data mining. *Data Min. Knowl. Discov.*, *15*, 21–28.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. (1980). The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys (CSUR)*, *12*(2), 213–253.

Fann, K. T. (1970). *Peirce's theory of abduction*. The Hague, The Netherlands: Mouton.

Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003, March). A survey of socially interactive robots. *Robotics and Autonomous Systems*, *42*(3–4), 143–166.

Hanheide, M., Gretton, C., Dearden, R., Hawes, N., Wyatt, J., Pronobis, A., ... Zender, H. (2011). Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In *Proceedings of IJCAI 2011.*

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

Hawes, N., Sloman, A., Wyatt, J., Zillich, M., Jacobsson, H., Kruijff, G.-J., ... Skočaj, D. (2007). Towards an integrated robot with multiple cognitive functions. In *AAAI* (p. 1548-1553).

Hawes, N., & Wyatt, J. (2010). Engineering intelligent information-processing systems with

CAST. *Adv. Eng. Inform.*, *24*(1), 27–39.

Hawes, N., Wyatt, J. L., Sridharan, M., Jacobsson, H., Dearden, R., Sloman, A., & Kruijff, G.-J. (2010, April). Architecture and representations. In H. I. Christensen, G.-J. M. Kruijff, & J. L. Wyatt (Eds.), *Cognitive Systems* (Vol. 8, pp. 51–93). Springer Berlin Heidelberg.

Hawes, N., Wyatt, J. L., Sridharan, M., Kopicki, M., Hongeng, S., Calvert, I., ... Zillich, M. (2010). The PlayMate system. In H. I. Christensen, G.-J. M. Kruijff, & J. L. Wyatt (Eds.), *Cognitive Systems* (Vol. 8, pp. 367–393). Springer.

Helmert, M. (2006). The fast downward planning system. *Journal of Artificial Intelligence Research*, *26*, 191–246.

Hobbs, J. R., Stickel, M. E., Appelt, D. E., & Martin, P. A. (1993). Interpretation as abduction. *Artificial Intelligence*, *63*(1–2), 69–142.

Janíček, M. (2012). Abductive reasoning for continual dialogue understanding. In M. Slavkovik & D. Lassiter (Eds.), *New directions in logic, language, and computation.* Springer.

Karaoguz, C., Rodemann, T., Wrede, B., & Goerick, C. (2012). Learning information acquisition for multi-tasking scenarios in dynamic environments. *IEEE TAMD*, *PP*(99), 1.

Kirstein, S., Denecke, A., Hasler, S., Wersing, H., Gross, H.-M., & Körner, E. (2009). A vision architecture for unconstrained and incremental learning of multiple categories. *Memetic Computing*, *1*, 291-304.

Kopicki, M. (2010). *Prediction learning in robotic manipulation* (Doctoral dissertation, University of Birmingham). Retrieved from `http://www.cs.bham.ac.uk/~msk/pdf/kopicki2010prediction.pdf`

Kristan, M., & Leonardis, A. (2013). Online discriminative kernel density estimator with gaussian kernels. *IEEE Trans. Syst. Man Cybern. B: Cybernetics*.

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

Kristan, M., Skočaj, D., & Leonardis, A. (2010). *Principles of discovering gaps in categorical knowledge* (Technical Report No. TR-LUVSS-03/10). University of Ljubljana, Faculty of Compuer and Information Science.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial intelligence*, *33*(1), 1–64.

Lallee, S., Pattacini, U., Lemaignan, S., Lenz, A., Melhuish, C., Natale, L., ... Dominey, P. (2012, September). Towards a platform-independent cooperative human robot interaction system: III an architecture for learning and executing actions and shared plans. *IEEE TAMD*, *4*(3), 239 –253.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lutkebohle, I., Peltason, J., Schillingmann, L., Wrede, B., Wachsmuth, S., Elbrechter, C., & Haschke, R. (2009). The curious robot-structuring interactive robot learning. In *Proceedings of ICRA 2009* (pp. 4156–4162).

Malfaz, M., Castro-Gonzalez, A., Barber, R., & Salichs, M. (2011, September). A biologically inspired architecture for an autonomous and social robot. *IEEE TAMD*, *3*(3), 232 –246.

Mason, M., & Lopes, M. (2011, March). Robot self-initiative and personalization by learning through repeated interactions. In *6th international conference on human-robot interaction (HRI)* (pp. 433 –440).

Meger, D., Forssén, P.-E., Lai, K., Helmer, S., McCann, S., Southey, T., ... Lowe, D. G. (2008). Curious george: An attentive semantic robot. *Robotics and Autonomous Systems*, *56*(6), 503–511.

Merrick, K. E. (2012, December). Intrinsic motivation and introspection in reinforcement learning. *IEEE TAMD*, *4*(4), 315 –329.

Otero, N., Saunders, J., Dautenhahn, K., & Nehaniv, C. L. (2008). Teaching robot companions: the role of scaffolding and event structuring. *Connection Science*, *20*(2-3), 111–134.

Oudeyer, P.-Y., Kaplan, F., & Hafner, V. (2007, April). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, *11*(2), 265 –286.

Perera, I., & Allen, J. F. (2013). SALL-E: situated agent for language learning. *AAAI Conference on Artificial Intelligence*, 1241–1247.

Poole, D. (1993). Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, *64*(1), 81–129.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Mach. Learn.*, *62*(1-2), 107–136.

Roy, D. K., & Pentland, A. P. (2002). Learning words from sights and sounds: a computational model. *Cognitive Science*, *26*(1), 113–146.

Salti, S., Tombari, F., & Di Stefano, L. (2014). SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, *125*, 251–264.

Salvi, G., Montesano, L., Bernardino, A., & Santos-Victor, J. (2012). Language bootstrapping: Learning word meanings from perception–action association. *IEEE Trans. Syst. Man Cybern. B, Part B: Cybernetics*, *42*(3), 660–671.

Schiebener, D., Morimoto, J., Asfour, T., & Ude, A. (2013, October). Integrating visual perception and manipulation for autonomous learning of object representations. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, *21*(5), 328–345.

Schmidhuber, J. (1991). Curious model-building control systems. In *Neural networks, 1991. 1991 IEEE international joint conference on* (pp. 1458–1463).

Sequeira, P., Melo, F. S., & Paiva, A. (2014). Learning by appraising: an emotion-based approach to intrinsic reward design. *Adaptive Behavior*, *22*(5), 330-349.

Steels, L., & Kaplan, F. (2000). AIBO's first words, the social learning of language and meaning. *Evolution of Communication*, *4*(1), 3–32.

Stickel, M. E. (1991). A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. *Annals of Mathematics and Artificial Intelligence*, *4*, 89–105.

Stone, M., & Thomason, R. H. (2003). Coordinating understanding and generation in an abductive approach to interpretation. In *Proceedings of DIABRUCK 2003.*

Sun, R. (2007). The importance of cognitive architectures: an analysis based on CLARION. *Journal of Experimental & Theoretical Artificial Intelligence*, *19*(2), 159-193.

Tellex, S., Thaker, P., Deits, R., Kollar, T., & Roy, N. (2012, July). Toward information theoretic human-robot dialog. In *Proceedings of robotics: Science and systems.* Sydney, Australia.

Tellex, S., Thaker, P., Joseph, J., & Roy, N. (2014). Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*, *94*(2), 151-167.

Thomaz, A. L., & Breazeal, C. (2008a). Experiments in socially guided exploration: lessons learned in building robots that learn with and without human teachers. *Connection Science*, *20*(2 3), 91–110.

Thomaz, A. L., & Breazeal, C. (2008b, April). Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, *172*(6–7), 716–737.

Vrečko, A., Janíček, M., Leonardis, A., & Skočaj, D. (2012). *Associating and merging multi-modal and multi-agent information in a cognitive system* (Technical Report No. TR-LUVSS-02/2012). University of Ljubljana, Faculty of Compuer and Information Science.

Vrečko, A., Leonardis, A., & Skočaj, D. (2012, November). Modeling binding and cross-modal learning in markov logic networks. *Neurocomputing*, *96*, 29–36.

Wohlkinger, W., Buchaca, A. A., Rusu, R., & Vincze, M. (2012). 3DNet: Large-Scale Object Class Recognition from CAD Models. In *Proceedings of ICRA 2012.*

Wyatt, J. L., Aydemir, A., Brenner, M., Hanheide, M., Hawes, N., Jensfelt, P., ... Skočaj, D. (2010, December). Self-understanding and self-extension: A systems and representational approach. *IEEE TAMD*, *2*(4), 282 – 303.

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

Zhou, K., Varadarajan, K. M., Zillich, M., & Vincze, M. (2011, Dec). Web mining driven semantic scene understanding and object localization. In *IEEE international conference on robotics and biomimetics (ROBIO).* Phuket, Thailand.

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