Towards Intelligent Social Robots: Current Advances in Cognitive Robotics

Amir Aly, Sascha Griffiths, Francesca Stramandinoli

Proceedings of the Full Day Workshop

Towards Intelligent Social Robots: Current Advances in Cognitive Robotics

in Conjunction with Humanoids 2015
South Korea
November 3, 2015

Amir Aly[1], Sascha Griffiths[2], Francesca Stramandinoli[3]

1- ENSTA ParisTech – France
2- Queen Mary University – England
3- Italian Institute of Technology – Italy

Towards Emerging Multimodal Cognitive Representations from Neural Self-Organization

German I. Parisi, Cornelius Weber and Stefan Wermter
Knowledge Technology Institute, Department of Informatics
University of Hamburg, Germany
{parisi,weber,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

***Abstract*—The integration of multisensory information plays a crucial role in autonomous robotics. In this work, we investigate how robust multimodal representations can naturally develop in a self-organized manner from co-occurring multisensory inputs. We propose a hierarchical learning architecture with growing self-organizing neural networks for learning human actions from audiovisual inputs. Associative links between unimodal representations are incrementally learned by a semi-supervised algorithm with bidirectional connectivity that takes into account inherent spatiotemporal dynamics of the input. Experiments on a dataset of 10 full-body actions show that our architecture is able to learn action-word mappings without the need of segmenting training samples for ground-truth labelling. Instead, multimodal representations of actions are obtained using the co-activation of action features from video sequences and labels from automatic speech recognition. Promising experimental results encourage the extension of our architecture in several directions.**

Keywords*—Human action recognition, multimodal integration, self-organizing networks.

I. Introduction

The ability to integrate information from different modalities for an efficient interaction with the environment is a fundamental feature of the brain. As humans, our daily perceptual experience is modulated by an array of sensors that convey different types of modalities such as vision, sound, touch, and movement [1]. Similarly, the integration of modalities conveyed by multiple sensors has been a paramount ingredient of autonomous robots. In this context, multisensory inputs must be represented and integrated in an appropriate way such that it results in a reliable cognitive experience aimed to trigger adequate behavioral responses. Multimodal cognitive representations have been shown to improve robustness in the context of action recognition and action-driven perception, learning by imitation, socially-aware agents, and natural human-robot interaction (HRI) [2].

An extensive number of computational models has been proposed that aimed to integrate audiovisual input (e.g. [3][4]). These approaches used unsupervised learning for generalizing visual properties of the environment (e.g. objects) and linking these representations with linguistic labels. However, action verbs do not label actions in the same way that nouns label objects [5]. While nouns generally refer to objects that can be perceived as distinct units, action words refer instead to spatiotemporal relations within events that may be performed in many different ways. In fact, action classification has been shown to be particularly challenging since it involves the processing of a huge amount of visual information to learn inherent spatiotemporal dependencies in the data. To tackle this issue, learning-based mechanisms have been typically used for generalizing a set of labelled training action samples and then predicting the labels of unseen samples (e.g. [15][16]). However, most of the well-established methods learn actions with a batch learning scheme, i.e. assuming that all the training samples are available at the training phase. An additional common assumption is that training samples, generally presented as a sequence of frames from a video, are well segmented so that ground-truth labels can be univocally assigned. Therefore, it is usually the case that raw data collected by sensors must undergo an intensive pre-processing pipeline before training a model. Such pre-processing stages are mainly performed manually, thereby hindering the automatic, continuous learning of actions from live video streams. Intuitively, this is not the case in nature.

Words for actions and events appear to be among children's earliest vocabulary [6]. A central question in the field of developmental learning has been how children first attach verbs to their referents. During their development, children have at their disposal a wide range of perceptual, social, and linguistic cues that they can use to attach a novel label to a novel referent [7]. Referential ambiguity of verbs could then be solved by children assuming that words map onto the action with most perceptual saliency in their environment. Recent experiments have shown that human infants are able to learn action-label mappings using cross-situational statistics, thus in the presence of piece-wise available ground-truth action labels [8]. Furthermore, action labels can be progressively learned and improved from social and linguistic cues so that novel words can be attached to existing visual representations. This hypothesis is supported by many neurophysiological studies evidencing strong links between the areas in the brain governing visual and language processing, and suggesting high levels of functional interaction of these areas during action learning and recognition [9].

In this work, we investigate how associative links between unimodal representations can naturally emerge from the co-occurrence of audiovisual stimuli. We show that it is possible to progressively learn congruent multimodal representations of human actions with neural self-organization using a special type of hierarchical connectivity. For this purpose, we extended our recently proposed neural architecture for the self-organizing integration of action cues [16] with an associative learning layer where action-word mappings emerge

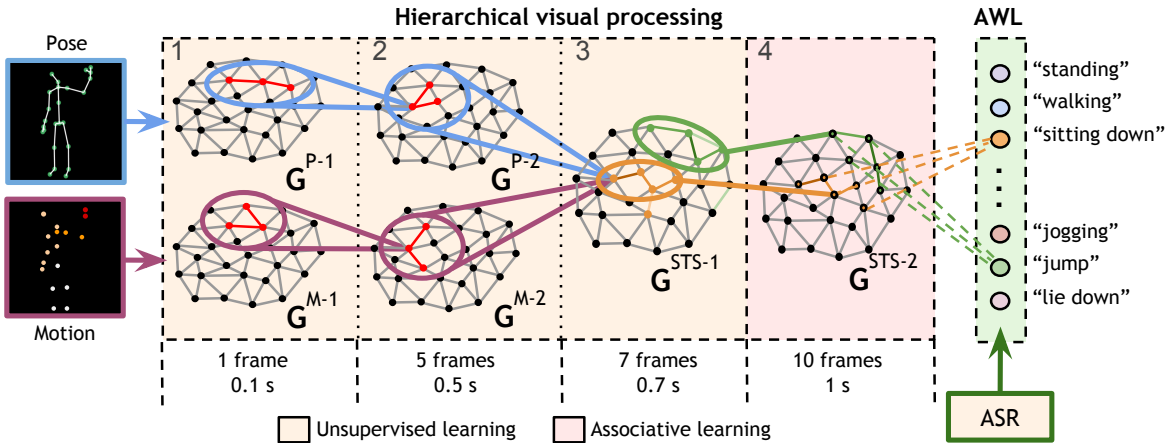


Fig. 1. Diagram of our learning architecture with GWR self-organizing networks and the number of frames (and seconds) required for hierarchical processing - Layers 1-3: parallel spatiotemporal clustering of visual features and self-organizing pose-motion integration (STS-1). Layer 4: associative learning for linking visual representations in STS-2 to the action words layer (AWL) obtained with automatic speech recognition (ASR).

from co-occurring audiovisual inputs using Hebbian-like learning [10]. We implement experience-dependent plasticity with the use of an incremental self-organizing network that employs neurobiologically-motivated habituation for stable learning [11]. The proposed architecture is novel in two main aspects: First, our learning mechanism does not require manual segmentation of training samples. Instead, spatiotemporal generalizations of actions are incrementally obtained and mapped to symbolic labels using the co-activation of audiovisual stimuli. This allows us to train the model in an online fashion with a semi-supervised learning scheme. Second, we propose a type of bidirectional inter-layer connectivity that takes into account the spatiotemporal dynamics of sequences so that symbolic labels are linked to temporally-ordered representations in the visual domain.

In Section II, we describe our hierarchical architecture with incremental self-organizing networks and hierarchical connectivity for multimodal integration. In Section III, we present our conducted experiments and compare our results with other approaches on a dataset of 10 actions using pose-motion cues as visual features and labels obtained from automatic speech recognition. In Section IV, we discuss on-going research efforts for the extension of our model in several directions.

II. PROPOSED METHOD

Our learning architecture consists of 4 hierarchically arranged layers and a symbolic layer of action words (Fig. 1). Layers 1 and 2 consist of a two-stream hierarchy for the processing of pose and motion features. One pathway processes body pose features while the other processes motion flow. The subsequent integration of pose-motion cues is carried out in Layer 4 (or STS-1) to provide movement dynamics in the joint feature space. The motivation underlying hierarchical learning is to obtain progressively specialized neurons coding spatiotemporal dependencies of the input, consistent with the assumption that the recognition of actions must be selective for temporal order. This is achieved by using trajectories of neuron activations from a network for the training of a higher-level network. A detailed description of Layers 1, 2, and 3 is provided by Parisi *et al.* [16].

From a neurobiological perspective, a large number of studies has shown that the superior temporal sulcus (STS) in the mammalian brain is the basis of an action-encoding network with neurons that are not only driven by the perception of dynamic human bodies, but also by audiovisual integration [13]. Therefore, the STS area is thought to be an associative learning device for linking different unimodal representations, accounting for the mapping of naturally occurring, highly correlated features such as shape, motion, and characteristic sound [14]. In our proposed architecture, we implement an associative learning network in Layer 4 (or STS-2) where action-word mappings are progressively learned from co-occurring audiovisual inputs using a self-organizing connectivity scheme.

A. Self-Organizing Hierarchical Learning

Our model consists of hierarchically-arranged Growing When Required (GWR) networks [11] that obtain progressively generalized representations of sensory inputs and learn inherent spatiotemporal dependencies. The GWR network is composed of a set of neurons with their associated weight vectors linked by a set of edges. The activity of a neuron is computed as a function of the distance between the input and its weight vector. During the training, the network dynamically changes its topological structure to better match the input space following competitive Hebbian learning [10].

Different from other incremental models of self-organization, GWR-based learning takes into account the number of times that a neuron has fired so that neurons that have fired frequently are trained less. The network implements a habituation counter $\eta(t) \in [0, 1]$ to express how frequently a neuron s has fired based on a simplified model of how the efficacy of an habituating synapse reduces over time [12]. The habituation counter is given by

$$\eta(s_t) = \eta_0 - \frac{S(t)}{\alpha} \cdot (1 - exp(-\alpha_t/\tau)), \qquad (1)$$

where $\eta(s_t)$ is the size of the firing rate for neuron s_t, η_0 is the resting value, $S(t)$ is the stimulus strength, and τ, α are constants that control the behaviour of the curve. A neuron

n is considered to be well trained when $\eta(n)$ is greater than a firing threshold η_T. This is in favour of training existing neurons before creating new ones. New nodes can be created any time if the activity of well-trained neurons is smaller than an activity threshold a_T. The GWR algorithm will then iterate over the training set until a given stop criterion is met, e.g. a maximum network size or a maximum number of iterations.

Hierarchical learning is carried out by training a higher-level network with neuron activation trajectories from a lower level network. These trajectories are obtained by computing the best-matching neuron of the input sequence with respect to the trained network with N neurons, so that a set of trajectories of length q is given by

$$\Omega^q(\mathbf{x}_i) = \{\mathbf{w}_{b(\mathbf{x}_i)}, \mathbf{w}_{b(\mathbf{x}_{i-1})}, ..., \mathbf{w}_{b(\mathbf{x}_{i-q+1})}\} \quad (2)$$

with $b(\mathbf{x}_i) = \arg\min_{j \in N} \|\mathbf{x}_i - \mathbf{w}_j\|$.

The STS-1 layer integrates pose-motion features by training the network with vectors of the form

$$\Psi = \{\Omega^q(\mathbf{X}), \Omega^q(\mathbf{Y})\}, \quad (3)$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the activation trajectories from the pose and motion pathways respectively. After STS-1 training is completed, each neuron will encode a sequence-selective prototype action segment.

B. GWR-based Associative Learning

For the higher layer STS-2, we extended the standard GWR algorithm with: 1) asymmetric neural connectivity based on Hebbian learning, and 2) semi-supervised labelling functions so that prototype neurons can be attached to symbolic labels during training. The detailed learning procedure for the creation and update of existing neurons is illustrated by Algorithm 1.

Local lateral connectivity in self-organizing networks is responsible for the correct formation of the topological map. We enhanced standard neuron connectivity by taking into account inherent temporal relations of the input, so that connections between neurons that are consecutively activated are strengthened. For this purpose, we define a connection strength function ρ that increases between activated neurons b_{t-1} and b_t at time $t-1$ and t respectively (Algorithm 1, Steps 6c and 7b). This type of connectivity scheme is asymmetric in the sense that $\rho(b_{t-1}, b_t)$ increases while $\rho(b_t, b_{t-1})$ remains unchanged, thereby fostering temporally-ordered representations of actions from neuron activation trajectories.

We extend the unsupervised GWR for semi-supervised learning so that action labels will be attached to prototype neurons during the training phase in an online fashion (Algorithm 1, Steps 6d and 7c). We implement a mechanism for label propagation that takes into account how well trained neurons are before propagating labels to their neighbours. For this purpose, we define two labelling functions: one for when a new neuron is created, and the other for when the neuron is updated. Provided that b_t is the index of the best-matching neuron and that ξ_t is the label of $\mathbf{x}_t$, and that we denote a missing label with -1, when a new neuron r_t is created, its label $\lambda(r_t)$ is assigned according to:

$$\gamma^{new}(b_t, \xi_t) = \begin{cases} \xi_t & \xi_t \neq -1 \\ \lambda(b_t) & \text{otherwise} \end{cases} \quad (4)$$

Algorithm 1 Semi-supervised Associative GWR

1: Create two random neurons with weights $\mathbf{w}_1$ and $\mathbf{w}_2$
2: Initialize an empty set of connections $E = \emptyset$.
3: At each iteration t, generate an input sample $\mathbf{x}_t$
4: For each neuron n, select the-best matching node and the second-best such that:
$b_t = \arg\min_{n \in A} \|\mathbf{x}_t - \mathbf{w}_n\|$
$s_t = \arg\min_{n \in A/\{b_t\}} \|\mathbf{x}_t - \mathbf{w}_n\|$
5: Create a connection if it does not exist
5a: $E = E \cup \{(b_t, s_t)\}$ and set age of E_{b_t,s_t} to 0.
6: If $(exp(-\|\mathbf{x}_t - \mathbf{w}_{b_t}\|) < a_T)$ and $(\eta(b_t) < f_T)$ then:
6a: Add a new neuron r_t between b_t and s_t with $\mathbf{w}_{r_t} = \kappa \cdot (\mathbf{w}_{s_t} + \mathbf{x}_t)$
6b: Create edges and remove old edge: $E = E \cup \{(r_t, b_t), (r_t, s_t)\}$ and $E = E/\{(b_t, s_t)\}$
6c: Connection strengths: $\rho_{(b_{t-1},r_t)} = 1, \rho_{(b_{t-1},b_t)} = 0$
6d: Initialize label: $\lambda(r_t) = \gamma^{new}(b_t, \xi_t)$
7: Else, i.e. no new neuron is added, update $\mathbf{w}_{b_t}$ and its neighbours i:
7a: $\Delta\mathbf{w}_{b_t} = \epsilon_b \cdot \eta(b_t) \cdot (\mathbf{x}_t - \mathbf{w}_{b_t})$ and $\Delta\mathbf{w}_{i_t} = \epsilon_n \cdot \eta(i) \cdot (\mathbf{x}_t - \mathbf{w}_{i_t})$,
with $0 < \epsilon_n < \epsilon_b < 1$ ents' request only, except (of course) you distribute your own 7b: Increase connection strength $\rho_{(b_{t-1},b_t)}$
7c: Update label: $\lambda(b_t) = \gamma^{update}(b_t, s_t, \xi_t)$
7d: Increment the age of all edges connected to b_t.
8: Reduce the firing counters η according to Eq. 1.
9: Remove all edges with ages larger than a_{max} and remove neurons without edges.
10: If the stop criterion is not met, go to step 3.

Provided that s_t is the index of the second best-matching neuron, the update labelling function for $\lambda(b_t)$ is defined as:

$$\gamma^{update}(b_t, s_t, \xi_t) = \begin{cases} \xi_t & \xi_t \neq -1 \\ \lambda(s_t) & (\xi_t = -1) \wedge (\eta(s_t) \geq \eta_T) \\ \lambda(b_t) & \text{otherwise} \end{cases} \quad (5)$$

This mechanism results in the correct propagation of labels so that labels attach to neurons based on the co-occurrence of audiovisual inputs, thereby avoiding the need of manual segmentation for ground-truth labelling.

C. Action-Word Mappings

During the learning in STS-2, unsupervised visual representations of actions are linked to symbolic action labels $\lambda_j \in L$, with L being the set of j possible words. Action words will then have a one-to-many relation with STS-2 neurons, i.e. neurons can be attached to only one label in L. It is possible that neurons change label during the learning phase based on the self-organizing process of label propagation. For clarity, we now refer to the symbolic connectivity layer of words as the "action words" layer (AWL).

The development of connections between STS-2 and AWL depends upon the co-activation of audiovisual inputs. More specifically, the connection between a STS-2 neuron and its symbolic label in AWL will be strengthened if the neuron is activated within a time window in which also the label is activated by an audio signal. In the case that no audio stimulus

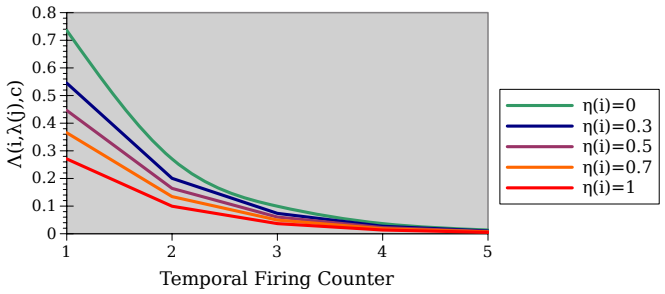


Fig. 2. Temporal strength function $\Lambda(i, \lambda_j, t) = 2 \cdot [exp(\eta(i) + c(\lambda_j, t))]^{-1}$ for different firing rates (y-axis) and sequence counters (x-axis). It can be seen how greater values are given to well-trained neurons activated at the beginning of the sequence.

occurs during the creation or adaptation of a STS-2 neuron, symbolic labels will instead be updated according to our semi-supervised label propagation rules (Eq. 4 and 5). This scheme takes into account the temporal order of activation in a given sequence of consecutively fired neurons. This is in favour of the generation of temporally-ordered trajectories generalizing one prototype action sequence. For a generic labelled neuron i fired at time t, its connection strength with the symbolic label λ_j becomes:

$$\Lambda(i, \lambda_j, t) = 2 \cdot [exp(\eta(i) + c(\lambda_j, t))]^{-1}, \quad (6)$$

where $c(\lambda_j, t)$ is the sequence counter and $exp(\eta(i) + c(\lambda_j, t))$ expresses the exponential relation between the firing counter of the neuron and its sequential order within the set of neuron activations with the same label. This function yields greater values for connections of well-trained nodes that activate at the beginning of a sequence. The counter $c(\lambda_j, t)$ will increase while $\lambda(b_t) = \lambda_j(t)$ and reset when this condition does not hold. The temporal strength function for different firing rates and sequence counters is depicted in Fig. 2 for a window of 5 neuron activations. A diagram of inter-layer connectivity between STS-1, STS-2, and AWL is shown in Fig. 3.

D. Action Word from Visual Recognition

At recognition time, we classify previously unseen video sequences to match one of the training actions. For this purpose, we define a recognition function $\varphi : \Omega \rightarrow \Lambda$ on the basis of a single-linkage strategy [19] such that each new trajectory sample ω_{new} from STS-1 is labelled with an action word $\lambda_j \in \Lambda$ associated to the STS-2 neuron $\mathbf{w}$ that minimizes the distance to the new sample:

$$\varphi(\omega_{new}) = \arg\min_{\lambda_j}(\arg\min_{\mathbf{w} \in N(\lambda_j)} \|\mathbf{w}_n - \omega_{new}\|). \quad (7)$$

The hierarchical flow is composed of 4 networks, with each subsequent network neuron encoding a window of 3 neurons from the previous one, with the exception of STS-2, which processes 4-neuron trajectories. Therefore, this classification algorithm returns a new action label every 10 samples (1 second of video operating at 10 frames per second). By applying a temporal sliding window scheme, we get a new action label for each frame.

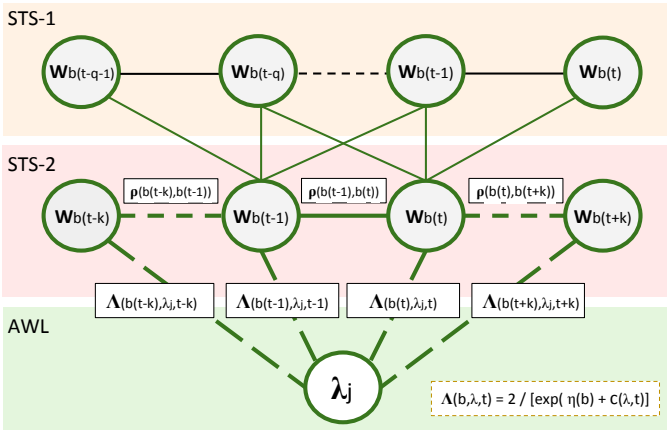


Fig. 3. Inter-layer connectivity scheme: Neurons in the STS-2 layer result from the hierarchical learning of STS-1 activation trajectories. STS-2 neurons use recurrent connection strength ρ to preserve temporal relations of the input. Connectivity between STS-2 and AWL emerges taking into account neuron firing rates and the order of activation.

E. Visual Sequence from Action Word

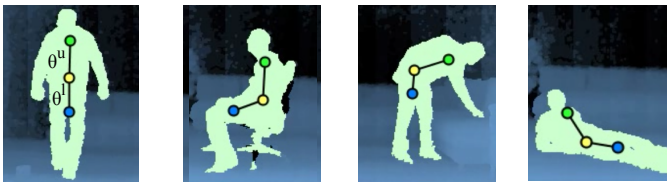
We use the strength function ρ to obtain prototype visual representations of actions from recognized action words. We expect that each action word will activate a trajectory that represents a prototype action sequence in the STS-2 layer. Therefore, after recognizing an action word λ_j from speech, the STS-2 neuron that maximizes Eq. 6 is selected as the first element of a sequence and used to generate temporally-ordered prototype representations of actions by recursive ρ-connectivity. This mechanism can be used in practice to assess how well the model has learned action dynamics and whether it has accounted for linking action words to visual representations.

III. EXPERIMENTS

We now present our experimental set-up and results on a dataset of full-body actions. In contrast to previous training procedures [15][16], for these experiments action samples from sequential frames were not manually segmented. Instead, action labels were recorded from speech so that action-word mappings of training samples resulted from co-occurring audiovisual inputs using our label propagation strategy. To evaluate our system, we compared new obtained results with recently reported results using GWR-based hierarchical processing with manual segmentation for ground-truth labelling [16].

A. Audiovisual Inputs

Our action dataset is composed of 10 full-body actions performed by 13 subjects [15]. Videos were captured in a home-like environment with a Kinect sensor installed 1,30 m above the ground. Depth maps were sampled with a VGA resolution of 640x480, an operation range from 0.8 to 3.5 m at 30 frames per second. The dataset contains the following actions: standing, walking, jogging, sitting, lying down, crawling, pick up, jump, fall down, and stand up. From the raw depth map sequences, 3D body joints were estimated on the basis of the tracking skeleton model and actions were represented by three body centroids (Fig. 4) as described in [15].



a) Walking b) Sitting c) Pick up object d) Lying down

Fig. 4. Representation of full-body movements from our action dataset [15]. We estimate three centroids: C_1 (green), C_2 (yellow) and C_1 (blue) for upper, middle and lower body respectively. The segment slopes θ^u and θ^l describe the posture in terms of the overall orientation of the upper and lower body.

For recording action labels, we used automatic speech recognition from Google's cloud-based ASR enhanced with domain-dependent post-processing [18]. The post-processor translates each sentence in the list of candidate sentences returned by the ASR service into a string of phonemes. To exploit the quality of the well-trained acoustic models employed by this service, the ASR hypothesis is converted to a phonemic representation employing a grapheme-to-phoneme converter. The word from a list of in-domain words is then selected as the most likely sentence. An advantage of this approach is the hard constraints of the results, as each possible result can be mapped to an expected action word. Reported experiments showed that the sentence list approach obtained the best performance for in-domain recognition with respect to other approaches on the TIMIT speech corpus[1] with a sentence-error-rate of 0.521. The audio recordings were performed by speaking the name of the action in a time window of 2 seconds during its execution, i.e. for each repetition in the case of jump, fall down, and stand up, and every 2 seconds for cyclic actions (standing, walking, jogging, sitting down, lying down, crawling). This approach has the advantage of assigning labels to continuous video streams without the manual segmentation of visual features.

B. Evaluation

For a fair comparison with previous results, we adopted similar feature extraction and evaluation schemes. We divided the data equally into training and test set, i.e., 30 sequences of 10 seconds for each periodic action (standing, walking, jogging, sitting, lying down, crawling) and 30 repetitions for each goal-oriented action (pick up object, jump, fall down, stand up). Both the training and the test sets contained data from all subjects. For GWR learning, we used the following training parameters: insertion threshold $a_T = 0.9$, learning rates $\epsilon_b = 0.3$, and $\epsilon_n = 0.006$, $\kappa = 0.5$, maximum age $a_{max} = 50$, firing counter parameters $\eta_0 = 1$, $\tau_b = 0.3$, $\tau_n = 0.1$, firing threshold $\eta_T = 0.01$. For a more detailed discussion on training parameters, please refer to Parisi *et al.* [16]

Experimental results showed that our new approach performs very well ($93,3\%$ average accuracy) with respect to our previous approach based on manual segmentation (94% average accuracy). The confusion matrix for the 10 actions is shown in Fig. 5 (with the rows of the matrix being the instances of actual actions and columns being the instances

[1]TIMIT Acoustic-Phonetic Continuous Speech Corpus: https://catalog.ldc.upenn.edu/LDC93S1

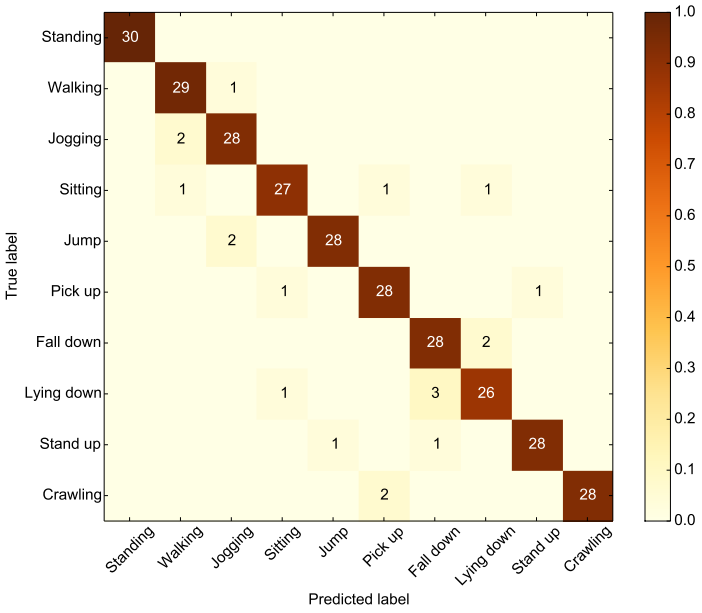


Fig. 5. Confusion matrix for our dataset of 10 actions. The average accuracy is 93,3%.

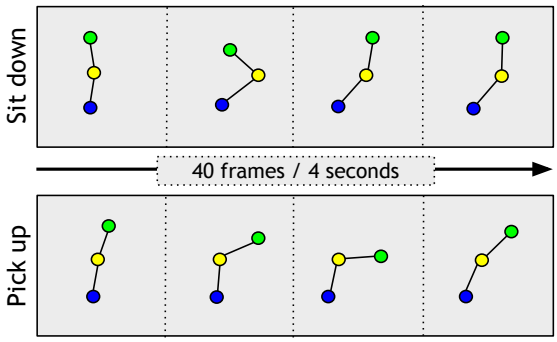


Fig. 6. Example of visual representations in STS-2 that maximize inter-layer connectivity for the actions "Sit down" and "Pick up" generated by speech recognition.

of predicted actions). These promising results encourage to extend our current neural architecture in several directions.

To have a qualitative idea of how well the associative layer has learned action dynamics, we extracted STS-2 neuron trajectories with the first neuron being activated by maximizing the temporal connection strength function (Eq. 6) and the subsequent 4 neurons obtained with ρ-connectivity. The visual representations of the actions "Sit down" and "Pick up" for a time window of 40 frames (4 seconds) are shown in Fig. 6, from which we can argue that the associative layer successfully learns temporally-ordered representations of input sequences.

IV. CONCLUSIONS AND FUTURE WORK

We presented a hierarchical neural architecture for action recognition from audiovisual inputs. In particular, we investigated how associative links between unimodal representations can emerge from the co-occurrence of multimodal stimuli in a self-organized manner. Experimental results on a dataset of 10 full-body actions have shown that our learning mechanism

does not require the manual segmentation of training samples for an accurate recognition. Instead, generalizations of action sequences are incrementally learned and mapped to symbolic labels using the co-activation of audiovisual inputs. For this purpose, we proposed a type of bidirectional, inter-layer connectivity that takes into account the spatiotemporal dynamics of action samples.

Similar to Vavrečka and Farkaš [4], we argue that the co-occurrence of sensory inputs is a sufficient source of information to create robust multimodal representations with the use of associative links between unimodal representations that can be progressively learned in an unsupervised fashion. Interestingly, our implementation with bidirectional action-to-word connections roughly resemble a phenomenon found in the human brain, i.e. spoken action words elicit receptive fields in the visual area [13]. In other words, visual representations of generalized actions can be activated in the absence of visual inputs, in this case from speech. We have shown that this property can be used in practice to assess how well the model has learned action dynamics.

This work represents the effort towards a more sophisticated learning-based model for the emergence of cognitive representations through the self-organizing development of associative links between different modalities. Current research work aims to leverage the proposed neural architecture in several directions. For instance, with our current implementation, we assume that labels are provided from speech during the training session for all the action samples. We are currently investigating the scenario in which labels are not always provided during training sessions, as it is also the case in nature. Several developmental studies have shown that human infants are able to learn action-label mappings using cross-situational statistics, thus in the presence of not always available ground-truth action labels [8]. Another limitation of our model is the use of domain-dependent ASR. In the future, we plan to avoid this constraint by accounting for learning new lexical features so that the action vocabulary can be dynamically extended during training sessions. For instance, it has been shown that lexical features can be learned using recursive self-organizing architectures [20][21]. Finally, we plan to evaluate our learning architecture with benchmark datasets using a greater number of body features. This is aimed to achieve more complex visual tasks such as the recognition of transitive actions.

ACKNOWLEDGMENT

This work was supported by the DAAD German Academic Exchange Service (Kz:A/13/94748) - Cognitive Assistive Systems Project.

REFERENCES

[1] B.E. Stein, T.R. Stanford, B.A. Rowland. The neural basis of multisensory integration in the midbrain: its organization and maturation. Hear Res 258(1–2):4–15, 2009.

[2] R. Kachouie, S. Sedighadeli, R. Khosla, and M-T Chu. Socially assistive robots in elderly care: a mixed-method systematic literature review. Int. J. Hum. Comput. Interact. 30:369–393, 2014.

[3] A.F. Morse, V.L. Benitez, T. Belpaeme, A. Cangelosi, and L. B. Smith. Posture Affects How Robots and Infants Map Words to Objects. PLoS ONE 10(3): e0116012. doi:10.1371/journal.pone.0116012, 2015.

[4] M. Vavrečka and I. Farkaš. A Multimodal Connectionist Architecture for Unsupervised Grounding of Spatial Language. Cogn Comput 6:101–112, 2014.

[5] D. Gentner. Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. In S. Kuczaj (Ed.), Language development: Language, thought, and culture 2:301–334, 1982.

[6] L. Bloom. The transition from infancy to language: Acquiring the power of expression. New York: Cambridge University Press, 1993.

[7] K. Hirsh-Pasek, R.M., Golinkoff, and G. Hollich. An emergentist coalition model for word learning. In R. M. Golinkoff et al. (Eds.), Becoming a word learner: A debate on lexical acquisition. New York: Oxford University Press, 2000.

[8] L. Smith and C. Yu. Infants rapidly learn word-referent mappings via cross-situational statistics. Cognition 106(3):1558–1568, 2008.

[9] F. Pulvermueller. Brain mechanisms linking language and action. Nature 6:576–582, 2005.

[10] T. Martinetz. Competitive Hebbian learning rule forms perfectly topology preserving maps. In: ICANN93 (Springer, Helderberg), pp. 427–434, 1993.

[11] S. Marsland, J. Shapiro, and U. Nehmzow. A self-organising network that grows when required. Neural Networks 15:1041–1058, 2002.

[12] J.C. Stanley. Computer simulation of a model of habituation. Nature 261:146–148, 1976.

[13] N.E Barraclough, D. Xiao, C.I. Baker, M.W. Oram, and D.I. Perrett. Integration of visual and auditory information by superior temporal sulcus neurons responsive to the sight of actions. J Cogn Neurosci 17(3):377–91, 2005.

[14] M.S. Beauchamp, K.E. Lee, B.D. Argall, and A. Martin. Integration of Auditory and Visual Information about Objects in Superior Temporal Sulcus. Neuron 41(5):809–823, 2004.

[15] G.I. Parisi, C. Weber, and S. Wermter. Human Action Recognition with Hierarchical Growing Neural Gas Learning. In Wermter, S., et al., editors. In: International Conference on Artificial Neural Networks (ICANN 2014), pp. 89–96, Hamburg, Germany, 2014.

[16] G.I. Parisi, C. Weber, and S. Wermter. Self-Organizing Neural Integration of Pose-Motion Features for Human Action Recognition. Frontiers in Neurobotics 9:3, 10.3389/fnbot.2015.00003, 2015.

[17] G.I. Parisi, G. I., F. v. Stosch, S. Magg, and S. Wermter. Learning Human Motion Feedback with Neural Self-Organization. In: International Joint Conference on Neural Networks (IJCNN), pp. 2973–2978, Killarney, Ireland, 2015.

[18] J. Twiefel, T. Baumann, S. Heinrich, and S. Wermter. Improving Domain-independent Cloud-based Speech Recognition with Domain-dependent Phonetic Post-processing. In: *IEEE Conf. on Artificial Intelligence (AAAI-14)*, pp. 1529–1535, Quebec, Canada, 2014.

[19] O. Beyer, and P. Cimiano. Online labelling strategies for growing neural gas. In: IDEAL11 (Norwich: Springer Berlin Heidelberg), pp. 76–83, 2011.

[20] M. Strickert and B. Hammer. Merge SOM for Temporal Data. Neurocomputing 64: 39–71, 2005.

[21] A. Andreakis, N. v. Hoyningen-Huene, and M. Beetz. Incremental unsupervised time series analysis using merge growing neural gas. In: Advances in Self-Organizing Maps, pp. 10–18. Springer, 2009.

Reusable Motivational Instruction Patterns for Socially Assistive Robots

Sebastian Schneider*, Michael Goerlich* and Franz Kummert*
*Applied Informatics
CITEC, Bielefeld University
Email: [sebschne,mgoerlic,franz]@techfak.uni-bielefeld.de

***Abstract*—Robots are increasingly tested in socially assistive scenarios. Future applications range from dieting, coaching, tutoring to autism therapy. In order to have a successful interaction with an artificial system, these systems need to have an interactional motivation model of how to assist users and encourage them to keep on with the task. In previous work, we have investigated how to build such a model for a specific scenario (e.g. indoor cycling). In this paper we want to show how to advance this model to be generalizable for other sport scenarios like rowing or bodyweight training. Therefore, we describe our framework for coordinating interaction scenarios with socially assistive robots.**

I. Introduction

Research in Socially Assistive Robotics (*SAR*) aims at designing scenarios, where robots instruct people during rehabilitation tasks, diet coaching or cognitive tasks [3, 6, 7]. Those scenarios and systems are often build from scratch and underlying interactional instruction patterns are hand-crafted for each scenario. This leads to recurring implementation of interaction structures for each scenario that are hard to compare across different systems or use cases. To the authors knowledge, few publications in SAR research exists which describe the architecture of their proposed SAR systems with a focus on how motivational feedback is generated and defined. Mead et. al. [9] describe in their paper an architecture for rehabilitation task practice in SAR. They describe a system offering different server and controllers managing the interaction between the robot and the human. However, it is not presented how the conversational feedback for motivating the user is designed during the rehabilitation tasks. In [5], Jayawarden et. al. propose a three layered architecture for rapid prototyping of SAR system and easy to use behavior description for subject matter experts (SME). However, the focus of their implementation was not on general motivational patterns robots can use for providing assistance. Therefore, the interaction and feedback provided by the robot is customized by the SME during an iterative end-user design process.

Because SARs usually provide hands-off support through verbal interaction and physical presence, the main challenge is to establish a common concept of the user's motivation and the compliance to interact with such systems. To evaluate and compare the effectiveness of the SARs (i.e. the encouraging support), needs to be modeled in interaction patterns that are reusable across different domains and applications. Reusability of common concepts and frameworks, which capture motivational support, in this domain can help researchers to measure the progress in this scientific field and to improve on previous established patterns. In previous work, we have targeted the application scenario of an robotic indoor cycling coach. During our investigations we have developed a motivational interaction model, which we have evaluated in an extended long-term study [14]. Currently, we are working on the generalizability of this model for different sport domains. We have advanced our previous implementation to ease to process of designing sport scenarios with robot assistance. Therefore, we pursue an integrated framework approach which targets four main advantages:

- Help non-expert programmers to implement robotic-assistance scenarios using a domain-specific language,
- use the same instruction patterns for each scenario,
- provide an easy to use configuration setup for the system to make decisions, and
- make components reusable.

We hope that the generalizability and reusability of our approach will help to build a toolbox which eases the process to explore new scenarios that require social assistance. The paper is organized as follows, first we will give a brief introduction of motivation as a key component for building SAR robots. Afterwards, we will explain our prior research efforts in this domain. In Section IV, we explain our current framework for designing SAR robotic scenarios and end our explanation in Section VII with an introduction of our current target scenarios. At last, we give a discussion and conclusion.

II. Motivation: A Key Component

In order to develop a common concept of motivational support for SARs, it is indispensable to identify the key components of motivation from the viewpoint of different disciplines. Motivational psychology discriminates two types of motivation: extrinsic and intrinsic motivation [4]. Extrinsic motivation itself can be divided into instrumental motivation, external self conception and goal internalization. Instrumental motivation influences the behavior of people based on a prospective external reward. External self conception is based on the perception of the ideal from one's personal role and the expectation of one's social surrounding. Goal internalization means that people make the corporate/institutional goals as

their own. In contrast, intrinsic motivation is divided into intrinsic process motivation, which means that someone is doing a task because of enjoying to do the task, and internal self conception, referring to behavioral change based on personal values and standards. Research has shown that intrinsic motivation is more effective for long-term interventions. Thus, many assistive systems make use of the theory of flow [1] for their task assistance and adapt the task difficulty to match the user's individual optimal challenge [2, 8]. Hence, motivation is often defined as a force which drives human behavior. This definition focuses on the internal states of an individual person. However, in socially assisted scenarios one main goal is also to collaboratively achieve a target. Therefore, also a sociological and linguistic perspective is important, which analyzes the different multi-modal cues during interactional processes. This means that some form of communication needs to be established which helps express one's desires and intentions. Therefore, future systems ideally also need to deal with wrong communication, need to have repair mechanisms and have a concept of when to trigger which kind of supportive feedback in a multi-modal manner in order to achieve a goal-oriented interaction [11].

III. Previous Work

In our previous work we have investigated the instructional structures and motivational strategies that human trainers incorporate into everyday workout (i.e. indoor cycling) in real world Human-Human Interaction. During field investigations, we recorded and observed the interaction between a coach and an athlete during indoor cycling session. The goal of this investigation was to identify some common interactional patterns or concepts of feedback and acknowledgment that coaches use to motivate and engage their athletes [13]. A qualitative analysis revealed a complex multimodal structure of motivation-relevant processes that are fine-grained and sequentially . This model had to be reduced to an interactive action-based motivation model due to the limitations of current robotic systems (see Fig. 1).

It captures the aspects of *preparation*, *instruction*, *acknowledgment*, *repair* and *feedback* (i.e. continuer-, encouraging-, positive-, end-oriented- feedback) in a systematic way for a single exercise instructions/movements. It has been implemented for a robotic-assisted indoor cycling scenario [14]. The states of the model where modeled as state charts. The transition between states where triggered based on assigned targets for each instruction and the resulting decision of specific decision servers (i.e. cycling with a specific cadence, power or posture). The implementation of this motivation model describes the instructional and motivational structure of **static movement patterns** (e.g. cycling with a target cadence) and **cyclic** repeating **movement patterns** (e.g. doing push ups, standing up - sitting down) in robotic assisted indoor cycling and has yet been tested only in this domain.

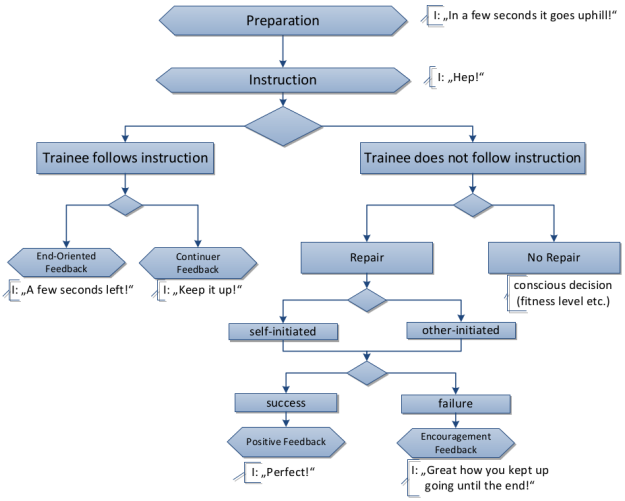
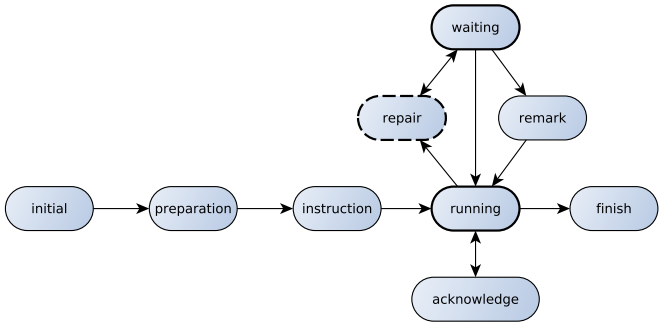
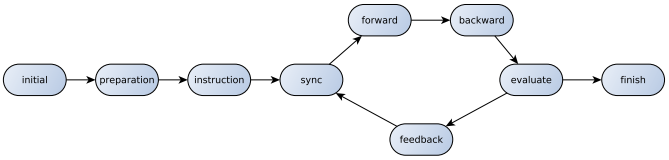


Fig. 1: Interactive action-based motivation model [14].



(a) Static Movement Instructions.



(b) Cyclic Movement Instructions.

Fig. 2: Generic instruction patterns for robot assisted sport scenarios.

IV. Towards a Reusable Motivational Instruction Model for Socially Assistive Robots

In our current work, we want to make these instruction models applicable for different sport domains as well as intuitive and reuseable for non-expert users. Our proposed scenario design for socially assistance is depicted in Figure 3. In the following, we explain and motivate the different concepts (e.g. scenario coordination and decision server) of our design.

A. Scenario Coordination

The scenario coordination is implemented using a domain specific language (DSL) which is automatically transformed

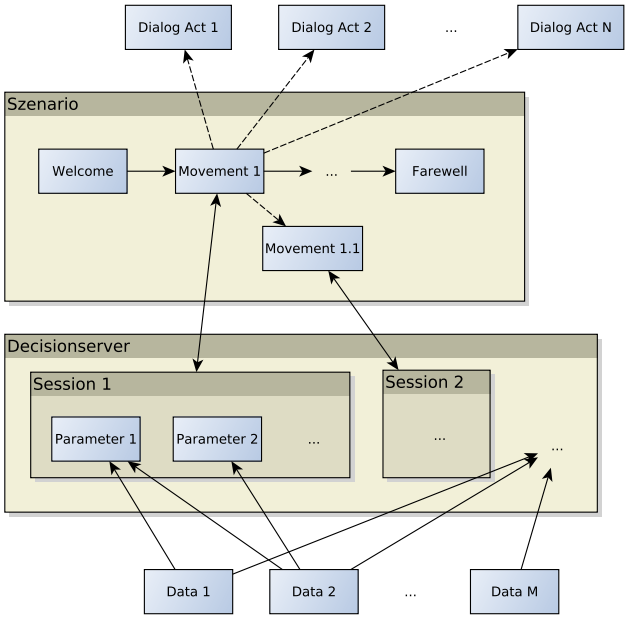


Fig. 3: Proposal for socially assistive scenario description.

to valid State-Chart XML[1] code (SCXML)[15]. State charts are commonly used to describe and coordinate the behavior of programs using events. Also the depicted movement patterns (see Fig. 2) are specified using the DSL. This specification includes the communication between different components in a distributed system and therefore simplifies the coordination (for details regarding the middleware see Section V of [10]). As tool, we use the Meta Programming System developed by JetBrains [2].

Each scenario is a state machine in which a number of different movements can be embedded and configured. Those movements represent the different exercises that a social robot can enquire a user to do. The movements are configured using the XML format. This configuration includes the actual dialog acts the robot produces during the different states of a movement as well as different targets (e.g. joint angle configuration of the user, speed or number of repetitions of exercises). Dialog acts can be any other state machine specified in our DSL. They can be simple text-to-speech acts, but also more complex dialog acts offered by a dialog system or even movements itself.

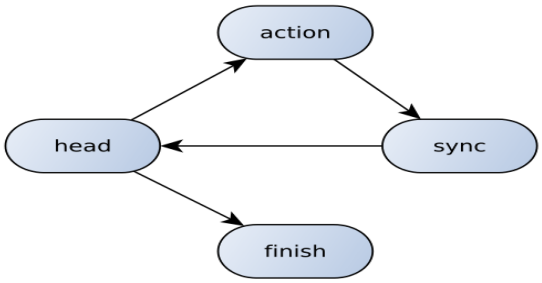
B. Hierarchical Instruction Patterns

Previously, each state was assigned one verbal instruction from the robot. However, for teaching or learning scenarios it is important that states can trigger interaction movements also. Therefore, we have introduced an hierarchical concept in the current design of our interaction models. This means that each state of a **static** or **cyclic movement** can be a movement itself (see Fig. 3, *Movement 1* initiates *Movement 1.1*). This approach allows to trigger an instructing movement in which the robot helps the person to reach a specific pose required for an exercise or to trigger a correcting exercise if the execution of the user was not adequate.

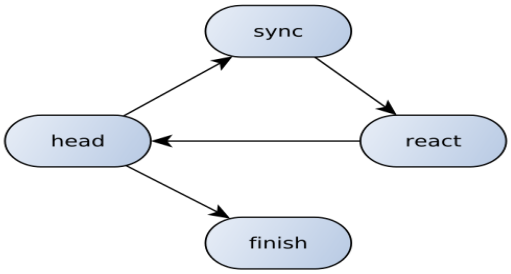
[1]http://www.w3.org/TR/scxml/
[2]https://www.jetbrains.com/mps/

C. Forward-Backward Cycle for Cyclic Instructions

Cyclic instructions are important for exercises where the user changes his/her position. Consider doing push ups. The user goes down towards the ground and up again which results in a complete push up cycle. For scenarios where the robot and the user are doing an exercise simultaneously together (see Figure 6) the interaction models needs to allow a synchronous execution. This requires the model to have states for going to different positions during a cyclic movement and to synchronize the action of the robot and the user at some point. This synchronization is achieved by a waiting task. During the wait task the decider verifies if the user has reached the desired position. If the user does not comply, the system will run into a time out and continues with the next execution of the cycle. However, there exist also exercises where the order of different states of the exercises are important one synchronization point is not sufficient. We have introduced the concept of **act-/react-actions** for this issue (see Figure 4). Those actions take place during the forward or backward states of the **cyclic movement**.



(a) The act-action.



(b) The react-action.

Fig. 4: Two possible actions for the *forwad-backward* states

1) Act-Instructions: During **react-actions**, the user is in charge of the tempo of the exercise execution and the robot follows the user‘s lead.

2) React-Instructions: In **act-actions** the robot is initating the exercises and waits for the user to follow.

V. Dynamic Decision Component

While the scenario coordination configures the interaction movements and executes them, the decision component receives the necessary information to make decisions during movement runtime (see Figure 3). Those decisions include recommendations to give a reparation, instruction, to praise the user or to terminate a movement. In the following, we describe the modular architecture of our decision system that is tailored to give designers of SAR scenarios the opportunity to build on a common framework for motivational support.

In general, decisions in assistive scenarios are based on some kind of data (e.g. strings, numbers, classification results). The way how decisions are made is inherently different between scenarios and implementation.

To give a main guidance for configuring decision systems and to add flexibility in the decision configuration for different scenario, we have implemented a **data-processing system**. This approach eases the process of deciding for different data types.

For each data type specific algorithms can be running which process the data and fulfill a special task. These algorithms are configurable and can be intertwined to solve more complex problems. The system defines components as well as input- and output-slots which can be connected.

A. Configurable Data-Processing Pipeline

There is a variety of components that can be used to configure a data-processing pipeline available. In the following we explain the current different categories that are used in our system:

1) datasource: Data sources create initial data, which are at the moment SimpleRSBDataSource. This source receives data from our used middelware Robotic Service Bus (RSB)[16]. The data source is configured using a scope, where the data events are expected, and an expected data type. Additional, there is the ManualDataSource which purpose is for *Unit-tests*.

2) transformations: Transformations transform, as expected, data into another data format. Since we are currently using a lot of skeleton information from the Kinect, which are represented as XML-strings in our system, we have a transformation component that deserializes the skeleton joints in 3D vector objects. Furthermore, we have a component that calculates the joint angle from three 3D vector objects. Hence, it is possible to compute each joint angle by configuration. Additionally, there is a transformation component which deserializes JSON data types. At last, we have a descriptive statistic components from the *Apache Commons math* library[3]. This components allows to compute a running mean or median from incoming numerical values.

3) deciders: Deciders transform in-slots to decision results. Currently, there is a decider for floating point numbers, which verifies that an incoming value is in a specific range and a decider for classification results which calculates the entropy and only passes on a decision if the entropy falls below a threshold. For example, those thresholds are certain joint angle configuration the user has to reach or a specified cadence he has to cycle. Since joint angle configurations are mostly the same for many people, we do not have to adapt the threshold. Regarding sports like indoor-cycling, we have run a fitness test with participants to determine their individual thresholds. In the future, those thresholds can also be adapted due to training adaption effects.

Furthermore, there are deciders that filter decisions of other components. This decider can be configured to pass on a negative decision only when it had been raised during a specific time period. At last, there exists the *one of many positive* deciders which checks whether one of many decisions are positive.

[3]https://commons.apache.org/proper/commons-math/

B. Local and Global Decisions

Each interaction session has a set of pairs of static and dynamic decision pipelines. One of these pairs reflects one exercise target of a movement (e.g. cadence of user during the cycling scenario). The static part is identical for each session and the dynamic part is distinct for one session. Furthermore, the static part is shared across sessions and usually does time dependent/consuming computations (e.g. average filter). The dynamic part always consists of at least one decider, which provides local decisions based on the results of the static part and the targets of the current movement.

Local decisions are represented as a *decision reason* which consists of the name of the parameter, the local decision, a timestamp and a boolean variable *good*, which indicates if a decision is negative or positive and reflects whether a goal is violated or not.

During one session all local decisions are collected into a *decision bag* (see Fig. 5). The *decision bag* is verified by the decider which then gives a guidance for a specific supportive behavior of the assistive system. Current implemented deciders are:

Simple Decider: The *simple decider* evaluates the *decision bag* for errors. Encountered errors are attached to the *decision reason*. If any errors are found, a *repair* advice will be send and the guidance is set to *failed*. If there is no error, an *acknowledge* is send.

Hierarchic Reaching: The *hierarchic reaching* allows to decide on multiple concurrent parameters. If one or many parameters are violated the *hierarchic reaching* can decide which parameter has priority.

Hierarchic Monitoring: The *hierarchic monitoring* decider is also an hierarchical decider. Instead of evaluating whether a target has been reached, it observers the specified parameters for a longer range of time.

At last, we have implemented components that *evaluate* the decisions. Those, are separated into *evaluation* strategies and *finishing* strategies. Those classes are decoupled from the decider because sometimes it is necessary to evaluate the state of the interaction due to different scenarios or contexts. The same goes for the finishing component, which can trigger the termination of a session, which in turn can trigger a

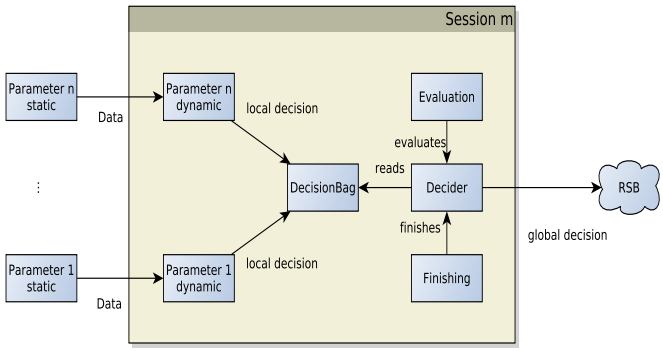


Fig. 5: Overview of the decision system.

supportive behavior. Currently we distinguish five different types of guidance: continue, repair, acknowledge, finished and failed. The details of the different guidance are:

continue: No reason for a change in the current situation.
repair: The known reasons make a reparation necessary.
acknowledge: The reasons favor a praise.
finished: The last known state was accurate.
failed: The last known state required a reparation.

The evaluation and finishing strategies are usually triggered after a certain amount of time has exceeded or after a threshold of specified events has been reached.

VI. Usage

So far we have described the different concepts that we see as building blocks for designing socially assistive robot scenarios. When a user wants to build a new scenario s/he can define an interaction flow using our provided IDE for developing RSB systems [10]. The user has to define at which points in the interaction what kind of movements should be triggered. If the user needs to define new movements, because there is no suitable movement configuration available, s/he can configure a new movement in XML format (for the limitation of the paper we will not include a detailed configuration) and define what the system should do depending on different measures. Those measures can be skeleton data, performance data from indoor bike, classification results or also new data provided from the user which are specific for a certain kind of scenario (e.g scores on a cognitive task [7, 12]). Depending on the type and goals of the intented scenario the user has to define what her/his parameters are. However, if certain parameter configuration already exist they can be easily included into a new configuration.

VII. Target Scenarios

In the previous section, we have introduced the different concepts and implementations that we have used to create a scenario coordination for SAR. We hypothesise that the described motivational concepts are universal across different scenarios or applications of SAR and that the set of functionalities is sufficient to many purpose.

To evaluate the generalizability of our proposed scenario coordination and motivational movement patterns, we have implemented three different robot-assisted sport scenarios. You can see examples of Human-Robot Interaction scenarios in Figure 6. In the following we briefly describe our current scenarios:

Indoor Cycling: During the indoor cycling scenario the robot is instructing the user to cycle at different speed or resistance and in different positions like standing, sitting or doing push ups on the bike. Each movement is finished after a specific time which is based on the length of the different songs that are played during the indoor cycling session. We have evaluated this scenario during a extended long-term study [14]

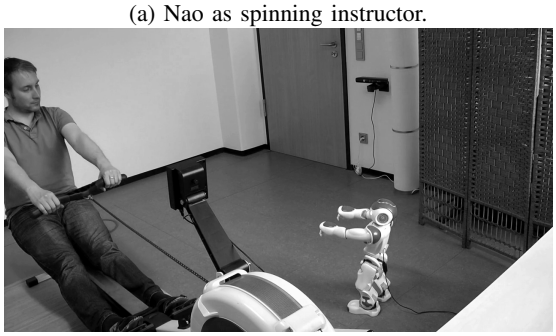
Rowing: In the rowing scenario the robot acts as a teacher explaining the user the different typical positions of a rowing stroke. It uses the concept of hierarchic reaching and repairs wrong stroke execution based on the following hierarchy: legs, back, arms. If one of the parameters is violated the system starts a movement which explains the correct execution of an exercise. We will compare this scenario against an interactive video which also explains the execution of a rowing stroke. As measure we will use the retention accuracy of how good participants remembered the steps of a rowing stroke after one week.

Body Weight Training: This scenarios aims at exploiting the embodiment of the robot. The robot and trainee do different exercises together (e.g. push ups, squats, lunges, etc.). We will implement different scenarios to test whether the user prefers robot initiated movements (see Fig. 4a) or self-initated movements (see Fig. 4b) and evaluate how different feedback strategies influence the assistive capabilities of the system.

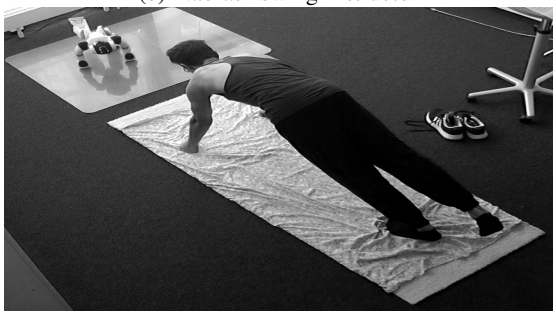
For all scenarios we use the same robot (i.e. Nao) in order to exclude effects due to the embodiment or appearance of the robot. Furthermore, we use the same decision system and scenario coordination as well as similar perceptive systems (skeleton tracking, heartrate, depth image of the user). We only needed to configure the explicit instructions and decision criteria which are unique for each interaction scenario.
Hence, we have acquired a state of the system where it is possible to reuse the same motivational model in all applications and use the same framework and implementations to create unique scenarios without worrying about implementational details. However, we need to evaluate whether the motivational model derived from indoor cycling scenarios is indeed applicable for other sport domains. Therefore, we will extensively test our target scenarios and evaluate the assistance in each scenario.

The implementation of these different scenarios results in a variety of different decision tasks and movement configuration. This different configurations are also reusable across scenarios and usable in new scenarios. We will work on building a set of configurations for movement and decision tasks that can be easily used when implementing a new SAR scenario.

(a) Nao as spinning instructor.



(b) Nao as rowing instructor



(c) Nao as bodyweight instructor.

Fig. 6: Different target scenarios using our proposed scenario coordination and movement patterns

At last, the concept of acknowledgement and reparation allows to easily compare different configurations for one scenario. The number of needed repairs can be used as a measurement to assess the effectiveness of the current configuration or classification system.

VIII. Conclusion

In this paper we have presented our proposed framework for designing and coordinating scenarios for socially assistive robot based on motivational instruction patterns. We have introduced the key concepts and components that will help to guide the design of scenarios across different application domains. Furthermore, we have presented three different sport scenarios where we already use our proposed framework. We hope that in the future, our approach can be used to better evaluate different scenarios using different robots which are based on the same underlying models.

In upcoming implementations, we also target to develop a domain specific language model for the configuration of movement and decision tasks. We hope this will enable non expert programmers to develop and configure instructions for new scenarios or enhancing and reusing existing implementations.

From a motivational perspective, we currently focused on motivation from a multi-modal instructional point of view. In the future, we will further work on the relation between the instructional model and the psychological model of motivation. Since every person needs different types of motivation strategies, it might also help to include a further layer in the current model. This layer can describe what kind of motivational instruction, in relation to extrinsic motivation, is appropriate for which kind of user.

Acknowledgments

This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology 'CITEC' (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

References

[1] M. Csikszentmihalyi. *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. Jossey-Bass, 25th anniversary edition, April 2000. ISBN 0787951404. URL http://www.worldcat.org/isbn/0787951404.

[2] Juan Fasola and Maja J Matarić. Socially assistive robot exercise coach: motivating older adults to engage in physical exercise. In *Experimental Robotics*, pages 463–479. Springer, 2013.

[3] David Feil-seifer and Maja J Matari. Defining socially assistive robotics. In *in Proc. IEEE International Conference on Rehabilitation Robotics (ICORR05*, pages 465–468, 2005.

[4] R. W. Scholl J. E. Barbuto. Motivation sources inventory: development and validation of new scales to measure an integrative taxonomy of motivation. In *Psychological Reports*, volume Vol. 82 (3), page 10111022, 1998.

[5] Chandimal Jayawardena, I-Han Kuo, Elizabeth Broadbent, and Bruce A MacDonald. Socially assistive robot healthbot: Design, implementation, and field trials. 2014.

[6] Cory D. Kidd and Cynthia Breazeal. Robots at home: Understanding long-term human-robot interaction. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 22-26, 2008, Acropolis Convention Center, Nice, France*, pages 3230–3235, 2008. doi: 10.1109/IROS.2008.4651113. URL http://dx.doi.org/10.1109/IROS.2008.4651113.

[7] Daniel Leyzberg, Samuel Spaulding, and Brian Scassellati. Personalizing robot tutors to individuals' learning differences. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, pages 423–430, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2658-2. doi: 10.1145/2559636.2559671. URL http://doi.acm.org/10.1145/2559636.2559671.

[8] Manuel Lopes, Benjamin Clement, Didier Roy, and Pierre-Yves Oudeyer. Multi-armed bandits for intelligent tutoring systems. *arXiv preprint arXiv:1310.3174*, 2013.

[9] Ross Mead, Eric Wade, Pierre Johnson, Aaron St. Clair, Shuya Chen, and Maja J. Mataric. An architecture for rehabilitation task practice in socially assistive human-robot interaction. In Carlo Alberto Avizzano and Emanuele Ruffaldi, editors, *RO-MAN*, pages 404–409. IEEE, 2010. ISBN 978-142447991-7. URL http://dblp.uni-trier.de/db/conf/ro-man/ro-man2010.html#MeadWJCCM10.

[10] Arne Nordmann, Sebastian Wrede, and Jochen Steil. Modeling of Movement Control Architectures based on Motion Primitives using Domain Specific Languages. In *International Conference on Automation and Robotics*, 2015.

[11] E. Schegloff. When "others" initiate repair. In *Applied Linguistics*, volume 21(2), page 205243, 2000.

[12] S. Schneider, I. Berger, N. Riether, S. Wrede, and B. Wrede. Effects of different robot interaction strategies during cognitive tasks. In *ICSR*, volume 7621 of *Lecture Notes in Computer Science*, pages 496–505. Springer, 2012.

[13] Luise Süssenbach. Interaction and motivation in fitness communication, 2011.

[14] Luise Süssenbach, Nina Riether, Sebastian Schneider, Ingmar Berger, Franz Kummert, Ingo Lütkebohle, and Karola Pitsch. A robot as fitness companion: towards an interactive action-based motivation model. 2014.

[15] W3C. State chart xml (scml): State machine notation for control abstraction, 2014.

[16] Johannes Wienke and Sebastian Wrede. A middleware for collaborative research in experimental robotics. In *SII2011*. IEEE, 2011.

Ubiquitous Semantics: Representing and Exploiting Knowledge, Geometry, and Language for Cognitive Robot Systems

Alexander Perzylo[1], Nikhil Somani[1], Stefan Profanter[1], Andre Gaschler[1]
Sascha Griffiths[2], Markus Rickert[1] and Alois Knoll[3]

***Abstract*— In this paper, we present an integrated approach to knowledge representation for cognitive robots. We combine knowledge about robot tasks, interaction objects including their geometric shapes, the environment, and natural language in a common ontological description. This description is based on the Web Ontology Language (OWL) and allows to automatically link and interpret these different kinds of information. Semantic descriptions are shared between object detection and pose estimation, task-level manipulation skills, and human-friendly interfaces.**

Through lifting the level of communication between the human operator and the robot system to an abstract level, we achieve more human-suitable interaction and thus a higher level of acceptance by the user. Furthermore, it increases the efficiency of communication.

The benefits of our approach are highlighted by examples from the domains of industrial assembly and service robotics.

I. Introduction

Knowledge representation for robotics is about connecting abstract representation with the "real world". Moreover, if a robot is deployed in an environment in which it will encounter humans or even other autonomous robots it will have to have flexible representations which allow an alignment of its own representations with those of the agents around it.

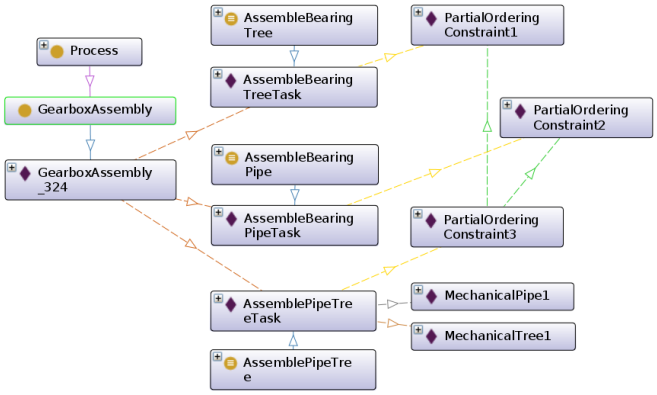
One can call this approach *ubiquitous semantics* which takes inspiration from the semantic web initiative. Using ontologies, one can tackle the problems which knowledge representation poses for modern robotics.

Ubiquitous semantics means that all relevant aspects of robot systems and their tasks are described in a way that preserves their inherent meaning. These semantic descriptions must be flexible and at a sufficiently generic level. This allows robots to share knowledge about how tasks are to be *performed and completed*. The descriptions are also flexible enough to describe the world in which the robot is moving but generic enough for a variety of environments and most importantly to allow for the *non-deterministic nature of the environments* in which robots are deployed, thus tackling the so-called "open world" problem. Also, such generic and flexible representations will be more amenable to the *plasticity of human communication*.

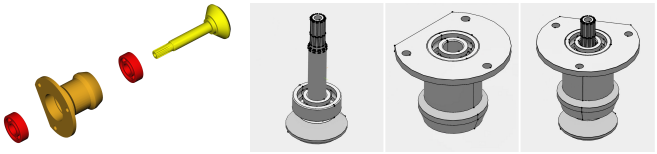
[1]Alexander Perzylo, Nikhil Somani, Andre Gaschler, Stefan Profanter, and Markus Rickert are with fortiss GmbH, Guerickestr. 25, 80805 München, Germany perzylo@fortiss.org

[2]Sascha Griffiths is with the Cognitive Science Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London, 10 Godward Square, London E1 4FZ, United Kingdom

Alois Knoll is with the Department of Informatics VI, Technische Universiät München, Boltzmannstr. 3, 85748 Garching, Germany



(a) Excerpt of semantic process description. Boxes containing a yellow circle represent classes, purple rhombi represent instances of these classes.



(b) Exploded view (c) Assembly steps

Fig. 1: Industrial assembly of four objects in three steps building the core part of a gearbox. Example from [1].

The remainder of the paper is structured in the following way. We will address the related work in the next section. This will be followed by a more general discussion of knowledge representation – specifically for robots. Against this background, we will discuss object detection, pose estimation, task execution, and human-friendly interfaces. We conclude with a few remarks on the general use of our ontology based knowledge framework.

II. Related Work

Related work applies concepts from knowledge representation [2], symbolic task planning [3], and planning for natural language dialogs [4].

Many modern approaches of knowledge representation in robotics have taken the semantic web initiative as a source of inspiration. Those approaches make use of ontologies to organize knowledge in autonomous and intelligent systems.

The RoboEarth initiative [5] makes use of this approach with the goal of achieving effective sharing of knowledge [2], data [6], and processing resources [7] among robots. This is often referred to as cloud robotics, and has established advantages regarding memory and processing limits.

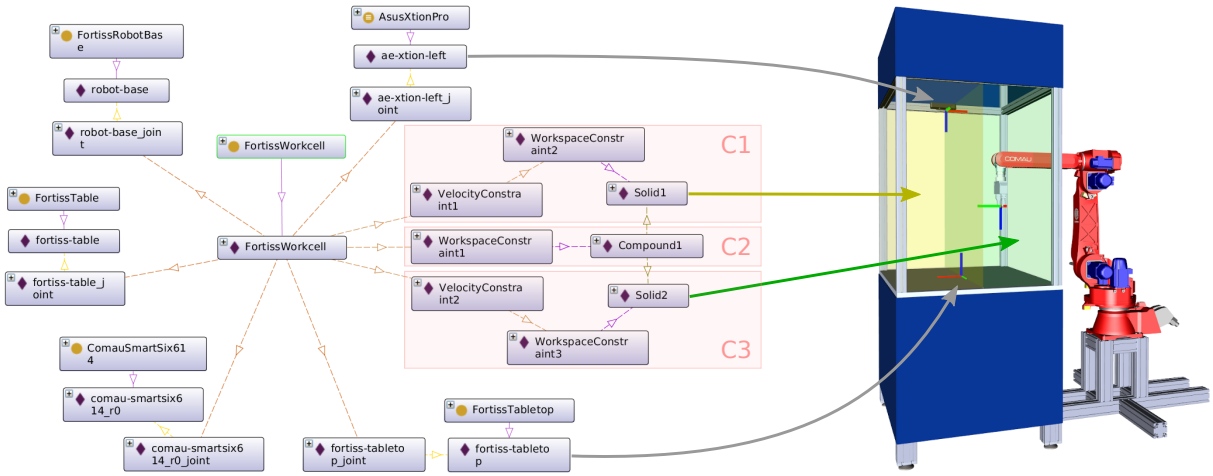


Fig. 2: A visualization of a robot workcell and an excerpt of the corresponding semantic description. Two velocity constraints (C1 and C3) and a workspace constraint (C2) have been specified.

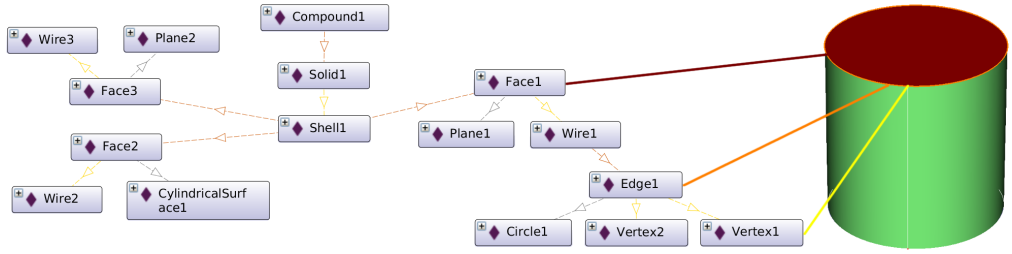


Fig. 3: Semantic description of a finite cylinder based on a boundary representation.

Additionally, models acquired by one robot can be re-used by another one.

There are other means by which robots can gain and apply knowledge. These can be categorized as "physical symbol grounding", "grounding words in action" and "social symbol grounding" [8].

III. Knowledge Representation

In order to endow robots with advanced cognitive capabilities, it is necessary to make all relevant aspects of their properties, tasks, and environment known to them. Encoding and interpreting knowledge about these different fields allows them to assess the applicability of their skills and to link their actions to a wider context.

In this section, we briefly summarize our methodology for semantically describing processes, related interaction objects, and their environment. We design a common description language based on the Web Ontology Language (OWL), which uses class taxonomies, instances of these classes, and properties for classes and instances.

A. Semantic process descriptions

Semantic process models are partially ordered sequences of tasks. Each type of task specifies its pre- and postconditions, and a set of parameters, of which some must be defined and others might be optional. An underspecified task can be fully parameterized through automatic reasoning, when a process model is assigned to a robot system, by combining the requirements of tasks with the capabilities of the selected system [1].

Fig. 1a depicts an excerpt of the semantic description of an industrial assembly process, which is visualized in Fig. 1b and Fig. 1c. It contains three tasks, i.e., *AssembleBearingTreeTask*, *AssembleBearingPipeTask*, and *AssemblePipeTreeTask*. Exemplarily, the associated object models for the *AssemblePipeTreeTask* are shown. The order of the tasks is given through *PartialOrderingConstraints*, which specify that the *AssemblePipeTreeTask* has to be executed after the other two tasks have been carried out.

B. Semantic environment description

Semantic environment descriptions encode the composition of physical entities of the real world, e.g., robots, tools, sensors, or tables, and abstract meta-information, e.g., available skills or environmental constraints [1].

The semantic description of the workcell in Fig. 2 specifies a robot, its base, a table, and an RGBD sensor. These entities are linked with the workcell instance *FortissWorkcell* through instances of type FixedJoint, e.g., *robot-base_joint*. The robot workspace is set to be constrained to the given cuboid (constraint C2), for which two subregions with different velocity limits have been defined (constraints C1 and C3).

C. Semantic object models

Next to basic object properties, e.g., type, name, weight, material, or bounding box, we are able to semantically

describe the geometric shape of objects using a boundary representation (BREP) [1], [9]. BREP preserves the exact mathematical models of contained curves and surfaces. This enables the system to define and interpret various geometric interrelational constraints, e.g., coincidence, concentricity, parallelity, etc., between two objects' vertices, edges, or faces [9], [10].

Fig. 3 shows the BREP-based semantic description of a finite cylinder's geometry. Selected correspondances between the visualization on the right and the ontological instances on the left are highlighted.

IV. Object Detection and Pose Estimation

In this section, we present an approach for shape-based object detection and pose estimation based on semantic descriptions of object models. This involves deep object models that include exact information about the geometric properties of the object. This approach allows for the detection of symmetrical objects whose pose are inherently underspecified. Knowledge about sensor noise and manufacturing tolerances can also be explicitly included in the pose estimation step [11].

A. Geometric constraints from primitive shape matching

The object is modeled as a set of primitive shapes P (e.g. planes, cylinders) based on its boundary representation (BREP). Each primitive shape $P_i \in P$ enforces a set of constraints $(\boldsymbol{C_{p_i}}, \boldsymbol{C_{n_i}})$ on the position and orientation of the object respectively, where each row of $\boldsymbol{C_{p_i}}$ and $\boldsymbol{C_{n_i}}$ contains a direction along which the constraint has been set.

A *complete set* of primitive shapes is defined as a set where the constraints fully specify the 3D position and orientation of the object. A *minimal set* of primitive shapes is defined as a set which is *complete* but removing any primitive shape from the set would render it incomplete.

Table II presents the list of supported geometric constraints between primitive shapes, where

$$\acute{\boldsymbol{p}}_2 = \boldsymbol{R}\boldsymbol{p}_2 + \boldsymbol{t}, \acute{\boldsymbol{p}}_{21} = \acute{\boldsymbol{p}}_2 - \boldsymbol{p}_1, \acute{\boldsymbol{n}}_2 = \boldsymbol{R}\boldsymbol{n}_2$$

1) Feature Vectors for Sets of Primitive Shapes: Correspondences between the scene and model shape primitives are obtained by matching feature vectors constructed from geometric properties of the primitive shapes. These feature vectors not only encode the geometric properties of the shapes, but also of the relations between the shapes (see Table I). Minimal sets of primitives from the scene point cloud are calculated during the pose estimation stage (see Section IV-B.2), and the distance between the feature vectors provides a metric for obtaining hypotheses of shape associations.

B. Constraint Processing for incomplete pose estimation

1) Detection of minimal and complete sets of primitives: The constraints (C_{p_i}, C_{n_i}) enforced by each primitive shape P_i are stacked into two matrices C_p and C_n (each having 3 columns). The constraints are *complete* if the matrices C_p and C_n both have rank 3. Fig. 4b shows an example of a complete set of primitive shapes.

TABLE I: Feature vectors for primitive shape sets

Primitive shape	**Feature Vector (*fv*)**
Inf. Plane	ϕ
Sphere	*radius*
Inf. Cylinder	*radius*
Plane+Plane	*fv(plane1), fv(plane2), angle(plane1_normal, plane2_normal), min_distance(plane1, plane2)*
Plane+Cylinder	*fv(cylinder), fv(plane), angle(plane_normal, cylinder_axis)*
Cylinder+Cylinder	*fv(cylinder1), fv(cylinder2), angle(cylinder1_axis, cylinder2_axis), min_distance(cylinder1, cylinder2)*
Plane+Plane+Cylinder	*fv(plane1, cylinder), fv(plane2, cylinder)*

Algorithm 1 Detecting object poses using RANSAC

1: **Input** : $[P_s, [[P_m]_{\min}]]$ (set of scene primitive shapes and minimal sets of model primitive shapes)
2: **Output** : $[\boldsymbol{T}, s_{\max}]$ (best pose estimate with score for detected object instance)
3: **forall** $P_i \in [P_m]_{\min}$
4: $s_{\max} \leftarrow 0$
5: compute shape matching hypothesis (H_i) using fv's, see Section IV-A.1
6: calculate transformation estimate $\boldsymbol{T_i}$ for H_i , see Section IV-B.2
7: compute score s_i for hypothesis H_i
8: **If** $s_i \geq \text{thresh}$ & $s_i > s_{\max}$
9: $\quad \boldsymbol{T} \leftarrow \boldsymbol{T_i}$
10: $\quad s_{\max} \leftarrow s_i$
11: **EndFor**

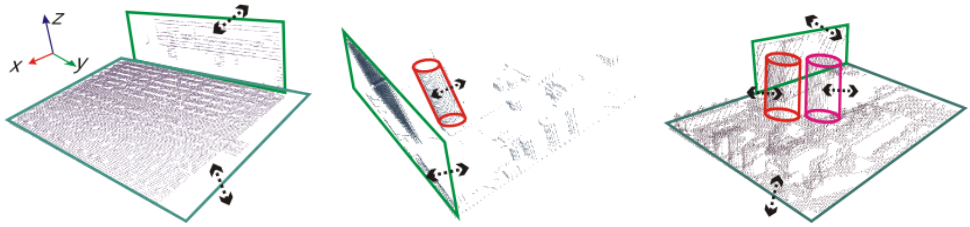
2) Constraint solving for pose estimation: The optimization is performed over transformations $\boldsymbol{T}$ that align the object model to the objects in the scene. The transformations are represented as $\triangle\boldsymbol{x} = (\boldsymbol{t}, \boldsymbol{r})$ where $\boldsymbol{t}$ is the translation and $\boldsymbol{r}$ is the rotation in axis angle representation.

The optimization function is the absolute value of the transformation, i.e., minimization of $\|\triangle\boldsymbol{x}\|_2$. The constraint functions g_i along with their lower and upper bounds ($\text{lb}(g_i)$, $\text{ub}(g_i)$) are obtained from the primitive shape matching constraints shown in Table II. The bounds $(d_{\min}, d_{\max})$ of the constraints can be used to incorporate the noise in sensor data or primitive shape fitting errors, as well as manufacturing uncertainties.

The resulting optimization problem is:

$$\begin{aligned} \arg\min_{\triangle\boldsymbol{x}} \quad & \|\triangle\boldsymbol{x}\|_2 \\ \text{subject to} \quad & \text{lb}(g_i) \leq g_i \leq \text{ub}(g_i),\ i = 1, \ldots, m. \end{aligned}$$

This set of equations is then solved using a non-linear least squares min-max solver (MA27) from [12] using the deterministic non-linear optimization utility from library Coin-OR (named IPOPT) [13]. If the constraints are complete, the pose is uniquely defined. Otherwise, the constraint solver returns one possible solution.



(a) (b) (c)

Fig. 4: Primitive shape groups for different views of an object. Using primitive shape sets, it can be calculated whether an object's pose can be fully estimated from a viewpoint. The arrows indicate the expected noise (qualitative only) in estimation of the primitive shape parameters. (a) the position of the object along y axis is not determined. In (b) and (c) the complete pose of the object can be estimated. (Note: The detected primitive shapes are highlighted for clarity.)

TABLE II: Summary of supported constraints between primitive shapes

Constraint (i)	**Cost Function** (g_i)	**Bounds** (lb, ub)	**Constrained Spaces**
Plane-Plane	$[\boldsymbol{n}_1^{\mathrm{T}}\acute{\boldsymbol{p}}_{21}; \acute{\boldsymbol{n}}_2^{\mathrm{T}}\boldsymbol{n}_1]$	lb : $[d_{\min}; a_{\min}]$ ub : $[d_{\max}; a_{\max}]$	$\boldsymbol{C_n}$: $[\boldsymbol{n}_{1_{\perp 1}}; \boldsymbol{n}_{1_{\perp 2}}]$ $\boldsymbol{C_p}$: $[\boldsymbol{n}_1]$
Cylinder-Cylinder	$[\|\acute{\boldsymbol{p}}_{21} - (\boldsymbol{n}_1^{\mathrm{T}}\acute{\boldsymbol{p}}_{21})\boldsymbol{n}_1\|_2^2; \acute{\boldsymbol{n}}_2^{\mathrm{T}}\boldsymbol{n}_1]$	lb : $[d_{\min}^2; a_{\min}]$ ub : $[d_{\max}^2; a_{\max}]$	$\boldsymbol{C_n}$: $[\boldsymbol{n}_{1_{\perp 1}}; \boldsymbol{n}_{1_{\perp 2}}]$ $\boldsymbol{C_p}$: $[\boldsymbol{n}_1]$
Sphere-Sphere	$[\acute{\boldsymbol{p}}_{21}]$	lb : $d_{\min}$ ub : $d_{\max}$	$\boldsymbol{C_n}$: $[\boldsymbol{n}_{1_{\perp 1}}; \boldsymbol{n}_{1_{\perp 2}}]$ $\boldsymbol{C_p}$: $[\boldsymbol{n}_1]$

3) RANSAC based constraint solving for pose estimation: A shape matching hypothesis H_i consists of a set of associations between primitive shape sets that can be computed by matching feature vectors (Section IV-A.1). An algorithm for pose estimation using RANSAC-like iterations on minimal sets of primitive shapes is described in Algorithm 1. For efficient hypothesis verification, we use the approach from [14] that utilizes geometric information from CAD models and primitive shape decomposition of scene point clouds.

V. Execution of Constraint-Based Robot Tasks

In order to execute a manipulation task in the workcell, the robot system's knowledge base is queried to obtain a set of task-specific geometric constraints. These constraints are then solved to obtain poses and residual null-spaces and to generate optimized robot trajectories [10].

In our task-level approach to robot execution, robot tasks are defined by geometric constraints that relate objects $\mathcal{O}$ and robot manipulators (including their tools) $\mathcal{R}$. A kinematic structure $\mathbf{R} \in \mathcal{R}$ is a tuple $(\mathrm{FK}, \mathbf{P})$, composed of a forward kinematic function FK that maps to the pose of its tool $\mathbb{R}^n \mapsto \mathrm{SE}(3)$ and a set of primitive shapes $\mathbf{P}$. A primitive shape $P \in \mathbf{P}$ may be one of the shapes defined in Sec. IV-A and serves as a useful reference for geometric relations, e.g. the grasp point of a parallel gripper. Analogous to kinematic structures, a manipulation object $\mathbf{O} \in \mathcal{O}$ is composed of a configuration and a set of primitive shapes, given by a tuple $(\mathbf{x} \in \mathrm{SE}(3), \mathbf{P})$.

A manipulation task is then defined by a set of constraints $\mathbf{C}$ that refer to primitive shapes of both kinematic structures and objects. Compared to the constraint resolution scheme in the object recognition component (Sec. IV-A), we perform a generic, iterative minimization of a cost function. For that, each constraint $C \in \mathbf{C}$ is represented by a cost function $\mathrm{Cost} : \mathrm{SE}(3) \times \mathrm{SE}(3) \mapsto \mathbb{R}^c$ that depends on the poses of two referenced shapes and returns a zero vector iff the constraint is fulfilled. To solve a given manipulation task, we minimize the stack of cost functions $\boldsymbol{q} \in \mathbb{R}^n \mapsto \mathbb{R}^c$ and obtain a valid robot pose $\boldsymbol{q}$. To ensure reliable convergence, cost functions are defined such that they are differentiable and reflect the correct number of c constrained degrees-of-freedom [10].

Many robot tasks in manufacturing and service domains pose constraints on only a few degrees-of-freedom, while the remaining degrees-of-freedom can be used to fulfill qualitative, lower-priority goals. Such goals may include the avoidance of singularities or joint limits, waypoints close to a previous one for shorter trajectories, or distance maximization from obstacles. When cost functions Cost allow computation of a full-rank Jacobian $\boldsymbol{J}$, we can compute the *null-space projection* matrix $\boldsymbol{N}$ of a task, $\boldsymbol{N}(\boldsymbol{q}) = \mathbf{1} - \boldsymbol{J}^{\dagger}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})$, where $\dagger$ denotes the pseudo-inverse. Projecting a lower-priority control signal onto $\boldsymbol{N}$ then allows null-space optimization of qualitative goals. As an example, the task of grasping a cylindrical object can semantically be defined by several coincidence constraints between a parallel gripper and the object. Based on these constraints, the robot will find a posture-optimized grasp along the rotational axes of the object.

VI. Human-Friendly Interfaces

We aim at reducing the complexity of interacting with robot systems. But, relying solely on semantic descriptions would only shift the required expertise for using such systems from the field of robotics to the field of knowledge

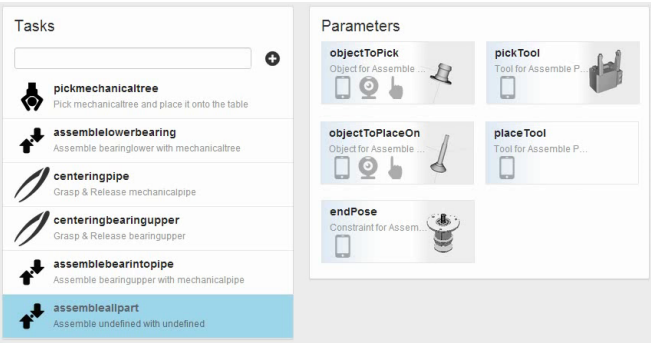


Fig. 5: Partial view of the intuitive interface which is used to program the robot and create or modify process descriptions.

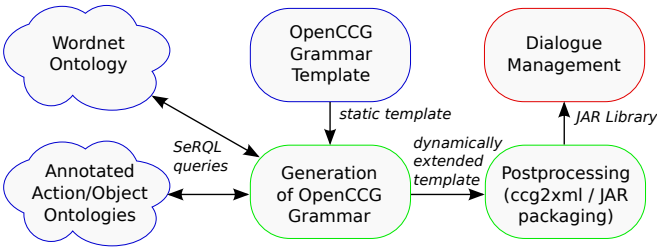


Fig. 6: Overview of configuration phase

engineering. Hence, we develop human-friendly interfaces, which act as a frontend to the semantic backbone.

A. Task-Level Programming Interface

The intuitive programming interface shown in Fig. 5 supports multiple input modalities: touchscreen, tracked 3D-pen, gestures, and speech [15]. By using these modalities during task-level programming, the user can define task parameters. We semantically describe modality and task parameter types, so that suitable modalities for certain parameter types can be automatically inferred and offered by the system [16].

For instance, the parameters *objectToPick* and *objectToPlaceOn* can be bound by selecting the desired object from a list, pointing at the object, or telling its name. This interface also supports the definition of assembly poses, grasp poses, and approach poses using geometric interrelational constraints [9], [10].

B. Natural Language Interface

This interface is not meant to support an open world dialog, but to instruct a robot system to perform a specific task. Interaction with our robot systems through natural language requires to map utterances to concepts in our ontologies, e.g., tasks and objects. We rely on a two-phases approach.

In the *configuration* phase, a human expert annotates the class taxonomies of tasks and objects with links to concepts in the Wordnet[1] ontology. As a second step in this phase, an OpenCCG[2] grammar is automatically generated [17], which serves as an input to our dialog component. The annotation has to be done only once for each type of task or object. The resulting grammar can be shared between all robots using our software framework. In the *runtime* phase, our dialog component uses the generated grammar to parse natural language input into a logical form, and to interpret it by mapping it back to concepts in the system's ontologies.

1) Configuration Phase: Natural language utterances can be ambiguous. As a result, a naïve one-to-one mapping of an instruction verb to a type of task would likely fail. Preferably, all synonyms for a given verb or noun should be considered, when trying to interpret a command. For this reason, we annotate the classes in the task and object ontologies with Wordnet synonym sets (synset). Task classes are annotated with verb synsets, object classes with noun synsets, and classes that serve as discriminating features with adjective synsets.

Fig. 7 exemplarily shows the annotation of a service robot's task description called *ServeBeverage*. It contains the *AnnotationProperty linkedSynset*, which links to a particular synset in the Wordnet ontology, i.e., *synset-serve-verb-6*.

```
Declaration(Class(re:ServeBeverage))
Declaration(AnnotationProperty(re:linkedSynset))
AnnotationAssertion(re:linkedSynset re:
    ServeBeverage <http://www.w3.org/2006/03/wn/
    wn20/instances#synset-serve-verb-6>)
```

Fig. 7: Excerpt of an annotated semantic task description in OWL functional syntax linking the task type with a synonym set of associated verbs.

The grammar generation process takes an OpenCCG grammar template[3] as an input. It contains the static parts of the grammar, i.e., functions, macros, and category definitions. The functions and macros are then used during the generation of the dynamic part of the grammar, e.g., to create the singular, singular third person, and plural forms of a verb. Furthermore, the template describes commonly used words which are not linked with concepts in our ontologies. For instance, definite and indefinite articles, prepositions, and pronouns. As a next step, the knowledge base is queried for all annotated task and object concepts, which results in a set of ontology concepts and their Wordnet synset annotations. The verbs, nouns, and adjectives from these synsets are then added to the grammar. An overview of the configuration phase in given in Fig. 6.

2) Runtime Phase: The OpenCCG grammar generated during the configuration phase is used by a dialog component to parse natural language utterances into a logical form. This representation is used to analyze a sentence's structure, and how the different parts are semantically related to each other, e.g., which noun is the subject of which verb. Starting from the logical form, the robot system has to determine, which task the human operator intends to be executed.

This is achieved by grounding the sentence's referents in

[1] https://wordnet.princeton.edu
[2] https://github.com/OpenCCG/openccg
[3] http://www6.in.tum.de/~perzylo/template.ccg

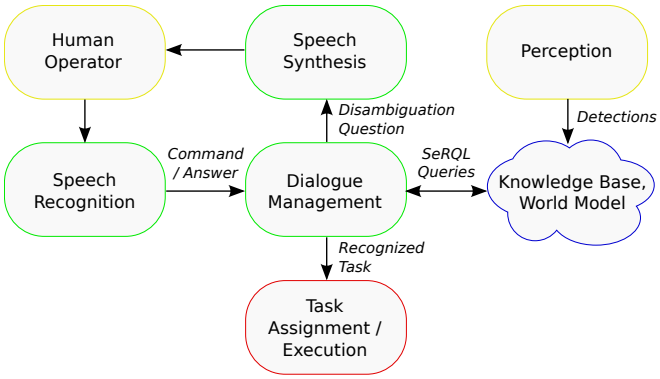


Fig. 8: Overview of runtime phase

the robot's knowledge base. Verb phrases are considered to correspond to a task that shall be executed. They have different numbers of associated noun or prepositional phrases, which form their arguments. They refer to objects the tasks have to be performed upon. Hence, each argument has to be grounded in the robot's knowlege base. The identification process first searches for all possible task candidates by matching the used verb with the synsets linked from the task concepts. This list is narrowed down by filtering out candidates, which require a different amount of arguments, or different types of arguments. If a single task could be identifed, it is selected for execution, otherwise a disambiguation dialog is initiated [17]. The runtime phase is summarized in Fig. 8.

VII. Conclusion

In this paper, we show how to specify and execute abstract process descriptions and their tasks, e.g., using geometric interrelational constraints between involved objects to define an assembly or grasp pose. The representation of deep object models, which are required to formulate such constraints on individual edges or faces, is based on the BREP formalism. It encodes the exact geometric properties of the objects' shapes. Using the knowledge on contained primitive shapes further improved the performance of our object detection and pose estimation.

In order to command the robot system through natural language, we automatically generate grammars to parse and map utterances to concepts in our ontological taxonomy of tasks and objects.

Having described all relevant aspects of a robot system and its tasks in a semantic way (ubiquitous semantics), the system can benefit from synergy effects created through linking the available information and reasoning about its implications.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287787 in the project SMErobotics, and FET grant agreement no. 611733 in the project ConCreTe.

References

[1] A. Perzylo, N. Somani, S. Profanter, M. Rickert, and A. Knoll, "Toward efficient robot teach-in and semantic process descriptions for small lot sizes," in *Proceedings of Robotics: Science and Systems (RSS), Workshop on Combining AI Reasoning and Cognitive Science with Robotics*, Rome, Italy, July 2015, http://youtu.be/B1Qu8Mt3WtQ.

[2] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz, "Representation and Exchange of Knowledge About Actions, Objects, and Environments in the RoboEarth Framework," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 643–651, July 2013.

[3] A. Gaschler, R. P. A. Petrick, M. Giuliani, M. Rickert, and A. Knoll, "KVP: A Knowledge of Volumes Approach to Robot Task Planning," in *IEEE/RSJ Intl Conf on Intelligent Robots and Systems (IROS)*, November 2013, pp. 202–208.

[4] M. E. Foster, A. Gaschler, M. Giuliani, A. Isard, M. Pateraki, and R. P. A. Petrick, "Two people walk into a bar: Dynamic multi-party social interaction with a robot agent," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI 2012)*, 2012.

[5] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Gálvez-López, K. Häussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schießle, M. Tenorth, O. Zweigle, and R. van de Molengraft, "RoboEarth," *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, June 2011.

[6] J. Elfring, S. van den Dries, M. J. G. van de Molengraft, and M. Steinbuch, "Semantic world modeling using probabilistic multiple hypothesis anchoring," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 95–105, Dec. 2012.

[7] D. Hunziker, M. Gajamohan, M. Waibel, and R. D'Andrea, "Rapyuta: The RoboEarth Cloud Engine," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 438–444.

[8] S. Coradeschi, A. Loutfi, and B. Wrede, "A short review of symbol grounding in robotic and intelligent systems," *KI-Künstliche Intelligenz*, vol. 27, no. 2, pp. 129–136, 2013.

[9] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[10] N. Somani, A. Gaschler, M. Rickert, A. Perzylo, and A. Knoll, "Constraint-based task programming with cad semantics: from intuitive specification to real-time control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[11] N. Somani, A. Perzylo, C. Cai, M. Rickert, and A. Knoll, "Object detection using boundary representations of primitive shapes," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, China, December 2015.

[12] "HSL. A collection of Fortran codes for large scale scientific computation," 2013. [Online]. Available: http://www.hsl.rl.ac.uk

[13] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[14] N. Somani, E. Dean-Leon, C. Cai, and A. Knoll, "Scene perception and recognition in industrial environments for human-robot interaction," in *9th International Symposium on Visual Computing*, July 2013.

[15] S. Profanter, A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "Analysis and semantic modeling of modality preferences in industrial human-robot interaction," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[16] A. Perzylo, N. Somani, S. Profanter, M. Rickert, and A. Knoll, "Multimodal binding of parameters for task-based robot programming based on semantic descriptions of modalities and parameter types," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Multimodal Semantics for Robotic Systems*, Hamburg, Germany, September 2015.

[17] A. Perzylo, S. Griffiths, R. Lafrenz, and A. Knoll, "Generating grammars for natural language understanding from knowledge about actions and objects," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, China, December 2015.

A Semantic Knowledge Base for Cognitive Robotics Manipulation

Elisa Tosello*, Zhengjie Fan† and Enrico Pagello‡
*‡Intelligent Autonomous Systems Lab (IAS-Lab)
Department of Information Engineering (DEI), University of Padova
Via Gradenigo 6/B, 35131 Padova, Italy
Email: *toselloe@dei.unipd.it, ‡epv@dei.unipd.it
†Big Data Team
Department of Big Data and IT Technology, China Mobile Research Institute (CMRI)
Innovation Building, No. 32, Xuanwumen West Avenue, Xicheng District, Beijing 100053, P.R. China
Email: fanzhengjie@chinamobile.com

***Abstract*—This paper presents a Semantic Knowledge Base that can be adopted as an information resource for autonomous cognitive robots performing manipulation tasks. Robots can use this database to save and share descriptions of learned tasks, detected objects, and explored environments. The Semantic Knowledge Base becomes: I) a data store for efficient and reliable execution of repeated tasks; II) a web-based repository for information exchange among robots.**

Compared to existing work the database does not only store the 3D CAD model of each object, it stores also its function and 3D geometric shape. A set of suitable manipulation poses is generated and stored according to the shape and the type of gripper. In this way, manipulation can be performed also on objects seen for the first time or for which only an incomplete 3D CAD model exists.

In this paper, we describe the ontology and the language adopted to define the knowledge base and to query it, together with the novel approach employed to solve the data sets interlinking problem related to the rescue and recovery of duplicate data. A sense-model-act framework is implemented to test the manipulation of an object which shape is in the database.

Keywords: Semantic Web, Cognitive Robotics, Manipulation, ROS

I. Introduction

One of today's robotics challenges is to improve robots' versatility: robots should be able to perceive and independently, safely and timely adapt to the constantly changing surroundings. The other challenge is to enhance efficiency: in order to intelligently act, robots should be able to learn from past experiences, errors, and actions performed by other agents (either humans or robots). These capabilities cannot arise from precompiled software routines. Robots must be able to reason like humans. For this purpose, a significant research program exists and is known under the name of *cognitive robotics*.

Over the years, cognitive robotics has been well investigated with regard to social robots. An example is the humanoid robot platform iCub [1]. With the advent of Industrie 4.0 [2], the field begins to be investigated by the industrial community. Smart Factories should be populated by manipulator robots able to flexibly adapt to constantly changing line configurations. A safe and productive human-robot interaction should be adopted in presence of ambiguous situations and tight workspaces. Robots can collaborate with humans or they can learn from humans, e.g., through a Learning From Demonstration framework similar to the one implemented in [3]. Moreover, multi-robots cooperative systems should speed up and improve the way to operate. A team of robots, in fact, can subdivide tasks according to their abilities, as suggested in the ontology proposed in [4].

This paper presents a Semantic Knowledge Base that can be adopted as an information resource for autonomous cognitive robots performing manipulation tasks. Robots that have learnt how to manipulate an object (e.g., exploring the workspace, observing a human demonstration or the actions performed by another robot), can save the acquired information in this base and share it with other robots. The purpose is not to formulate a *cognitivist* precoded symbolic representation of the workspace, but to create a knowledge base able to improve the *co-determination* of *emergent systems*.

Traditional cognitive modeling approaches involve symbolic coding schemes as a means for depicting the world. This symbolic representation originates a designer-dependent action domain [5] that is successful if the system acts under the conditions specified by descriptors; otherwise, a *semantic gap* [6] between perception and possible interpretation follows and must be bridged implementing ad-hoc frameworks [7]. Codetermination is the solution. It means that the agent constructs its reality (its world) as a result of its operations in the world: intelligently acting is functionally-dependent on the interaction between perception and cognition.

Focusing on manipulation, the robot should be able to perceive and explore the surrounding workspace, e.g. combining vision sensors with navigation capabilities, in order to detect objects to be manipulated, obstacles to be avoided, or actions performed by human teachers; it has to compute or learn the path that allows the manipulation of objects without collision; it has to learn how to approach the object (e.g., the grasp pose and grasp force), e.g. combining a trial-and-error reinforcement learning with a matching phase that compares the achieved results with the desired ones. It should

be able to store descriptions of learned tasks, detected objects, and explored environments creating a knowledge base that facilitates the replication of actions by the same robot or by different robots with the same assignment. Creating a Knowledge Base that can be adopted as information resource become fundamental to create intelligent cognitive robots.

The rest of the paper is organized as follows. Section II gives an overview of existing work on using a Semantic Knowledge Base, and our general approach to its design. Section III describes in details our implementation with regards to its instances, the language used to define and query it, and the interlink algorithm adopted to avoid replicas. Section IV describes the experiment set up to prove the good functioning of the system. Section V contains conclusions and future work. Authors discuss about how the adoption of this system can face robotics challenges such as knowledge acquisition, learning from previous tasks, and dissemination of knowledge to other robots.

II. Related Work

Many existing works aim to define and populate a robotics knowledge base. The Semantic Object Maps (SOMs) of Rusu at al. [8], its extended version (SOM$^+$s) presented by Pangercic at al. [9], the Columbia Grasp dataset [10], the MIT KIT object dataset [11], and the Willow Garage Household Objects Database [12] are available on line. KnowRob [13], the knowledge base of RoboEarth [14], is the most widespread. The Semantic Object Maps only stores information about objects in the environment, including 3D CAD models of objects, their position and orientation, their appearance, and function. The others also give information about grasp poses and can be used to evaluate different aspects of grasping algorithms, including grasp stability [15], robust grasping [16] and scene understanding [17]. The Household object database is a *simple* SQL database. All other approaches aim to make robots autonomous and able to store and share data without human intervention. Hence, as stated in [18] for the RoboEarth language, information is represented in a machine-understandable format, the same format required by the Semantic Web [19], in which computers exchange information between each other. The meaning of the content needs to be represented explicitly in terms of separating logical axioms that a computer can understand. These logical axioms need to be well-defined, for example in an ontology.

Similar to RoboEarth, our Semantic Knowledge Base is defined by an ontology and provides an interface to the open-source Robot Operating System (ROS) [20] that guarantees its reusability. The Base defines a semantic representation language for actions, objects, and environment. It contains 3D models of objects and their functions, but these models are not essential to manipulate the queried objects. Existing databases stores objects as triangular meshes. Stored items are of high quality, but object creation required either a lot of manual work or expensive scanning equipment. In order to save time and money, RoboEarth models objects as 3D colored point clouds. [21]. Anyway, each object model still consists of several recordings from different points of view. In a real world scenario I) it is difficult to reconstruct the 3D model of an object seen for the first time II) manipulation can be performed independet of the 3D model of the object. For these reasons, without loss of information, the proposed database models objects as a set of basic 3D geometric shapes, such as Sphere, Cylinder, Cube, Cone, and etc. Manipulation configurations are generated according to these shapes. In this way, it is possible to manipulate known objects (objects which 3D model is saved in the Cloud), objects according to their functions (objects which functon is saved in the Cloud), novel objects (objects which shape is saved in the Cloud).

The knowledge base can be used either by service robots or industrial manipulators, either by autonomous robots exploring their surrounding or robots learning from demonstrations of other agents (either human or robotic).

Instances of the ontological database are interlinked through a novel interlinking algorithm [22]. As stated in [23], large datasets collected from distributed sources are often dirty with erroneous, duplicated, or corrupted data. The adopted interlinking algorithm finds the interlink pattern of two data sets applying two machine learning methods, the K-medoids [24] and the Version Space [25]. This algorithm largely reduces the computation of comparing instances with respect to the commonly used manually interlinking.

III. The Semantic Knowledge Base

The implemented Semantic Knowledge Base contains descriptions of manipulation actions (grasp, push, place), objects (3D models and shapes), and environments (trajectories already performed). A single database encodes the information and collects the data provided by different robots with different sensing and gripping capabilities. The robot can query the knowledge base to retrieve known information or it can save new information and share it with other robots. To improve re-usability, the knowledge base is fully integrated into ROS. The database and the ontology are accessible via the Web and can easily be extended by users (see Figure 1).

A. Classes and instances

Let a robot, equipped with a gripper and a vision system, stand in the scene (real or simulated environment). The database encodes the scene as follows.

- **Automaton**: stores automata models. Every automaton is composed of a tuple <**Robot**, **Gripper**, **Sensor**>. The system has a self-model consisting of an XML file describing its kinematic structure, and a semantic model describing the meaning of the system's parts (e.g., robot, gripper, vision sensor). The Universal Robot Description File (URDF) format is used to represent the kinematic model, and the Semantic Robot Description Language (SRDL) [26] is used to describe every component and its capabilities, matching them against the requirements specified for the action space;
- **Robot**: models some robots (robot joints and limits);

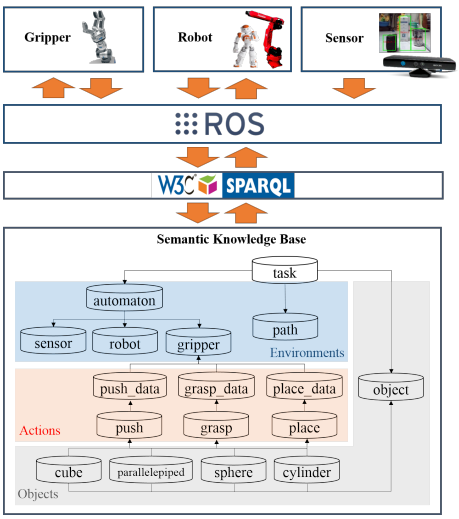


Fig. 1. The Semantic Knowledge Base

- **Gripper**: models some robotics end-effectors. It contains information about gripper joints and their limits. The authors distinguish the **Robot** class from the **Gripper** one because a user can attached a single gripper to different types of robotic arms. The same is true in reverse. The combination of an arm (instance of **Robot**) and a gripper (instance of **Gripper**) forms an instance of **Automation**. If a robot incorporates a gripper, e.g., the humanoid Aldebaran NAO[1], then the instance *NAO* of the class **Robot** will not be linked to any instance of **Gripper**. In this case, **Automation** will consist only of **Robot**;
- **Sensor**: models some robotics sensors. E.g., the Kinect vision sensor [27].

Every automaton in the scene can map its workspace and construct a 2D or 3D map of its surrounding. The map includes the trajectories performed.

- **Path**: contains Cartesian pose points sampled by a motion planning algorithm to lead a robot, or its end-effector, from a start to an end configuration.

Also the description of encountered objects is available.

- **Object**: describes 3D object surface models. It contains the list of object's meshes. Every object has a function (e.g., drink) and a pose [x, y, z]. It is connected to its shape. Every object is associated with a shape, but a shape could be correlated with no item. This feature guarantees the chance to model unknown objects;
- **Sphere**: represents spheres. It contains elements such as the $radius$ of the sphere;
- **Cylinder**: represents cylinders ($radius$, $height$);
- **Cube**: represents cubes ($side$);
- **Parallelepiped**: represents parallelepipeds (*height*, *width*, *depth*).

Currently, detected objects are mapped only into the four above-mentioned 3D basic shapes (Sphere, Cylinder, Cube, Parallelepiped). If a computed shape does not perfectly match with its corresponding object, the manipulation task will still end well thanks to the application of an *Effect Evaluator* which will refine the poses of the end-effector (see Section IV). Please note that the database is available on-line and can be extended by users.

This information allows robots to efficiently manipulate already manipulated objects or navigate already navigated environments. In detail, a task is assigned. It asks the automaton for the fulfillment of a manipulation action on an object. The knowledge base provides support to execute commands like *'Grasp a cup!'*, *'Grasp something suitable for drinking!'*, *'Grasp the object at pose [x, y, z]!'*, that means to manipulate a known object, an object for which the function is known, a novel object. The first query needs a 3D description of the object and its recognition, the second one requires an ontology linking objects with their function, the last one involves the detection of new objects and a way to represent the retrieved information into the dataset. The best way we found is associating the object with its shape and generating the manipulation action according to this shape.

- **Task**: Contains the list of tasks. Every instance models the action to be performed and the object to be manipulated. Information about the initial and final pose of the object are included, together with the time limit under which the action must be completed. For every task, its outcome is reported as a string (*completed, error, running, to_run*) describing the final state of the manipulation.

Studies have demonstrated that: I) placing the arm at a certain distance in front of the object before acting improves actions; II) humans typically simplify the manipulation task by selecting only one among different prehensile postures based on the object geometry [28]. Following this approach, the MoveIt! Simple Grasps tool[2] implemented by Dave T. Coleman provides the following instances:

- **Grasp**: contains the safety distance to be kept during the pre-grasp phase and the grasp poses [x, y, z, $roll$, $pitch$, yaw] of the gripper;
- **Grasp_data**: contains the joints configuration that the gripper has to maintain during the pre-grasp and grasp phases.

We extended the MoveIt! Simple Grasps tool and our implementation provides the **Grasp** and **Grasp_data** instances, and besides, the following istances:

- **Push**: as **Grasp**;
- **Push_data**: as **Grasp_data** but with different joints configurations;
- **Place**: contains the place poses [x, y, z, $roll$, $pitch$, yaw] of the gripper and the distance that the gripper has to mantain after the place;
- **Place_data**: contains the gripper's joints configurations. The grasp configuration becomes the place one and the pre-grasp configuration becomes the post-place one.

[1]Aldebaran NAO robot: https://www.aldebaran.com/en/humanoid-robot/nao-robot

[2]MoveIt! Simple Grasps tool: https://github.com/davetcoleman/moveit_simple_grasps

The actual organization of the knowledge base allows the definition of an actions' hierarchy that is able to generate complex actions by composing simple actions. Examples follow:

- **Put**: **Grasp** ∩ **Place**.

B. The ontology

As in [4], we first have to choose the appropriate Semantic Web language to describe the information provided by the database. We compared Extensible Markup Language (XML)[3], Resource Description Format (RDF)[4] and Web Ontology Language (OWL)[5]. XML lets the definition of a format, it is verbose and data exchange requires a set of basic rules to allow different systems to communicate and understand each other. RDF is used to define a model and it does not need to be redefined when new knowledge should be stated: its schema stays the same. If we want to define an ontology, we do not have to define a message format. We have to define a knowledge representation, naming and defining the types, properties, and interrelationships of the entities of a particular domain [29]. For this reason, we selected the union of RDF and OWL, namely OWL Full. RDF is used to define the structure of the data, OWL adds semantics to the schema and allows the user to specify relationships among the data. OWL Full allows an ontology to augment the meaning of the pre-defined RDF vocabulary guaranteeing the maximum expressiveness of OWL and the syntactic freedom of RDF. Indeed, OWL is adopted by the Word Wide Web Consortium (W3C)[6] and it is the representation language used by the IEEE Robotics and Automation Society (RAS)'s Ontologies for Robotics and Automation (ORA) Working Group [30], [31], [32].

The ontology describes the relationship between the defined classes and their attributes. For example, for each entity of the class **Object** it defines the properties *shape* and *function*. It associates the shapes to the suitable manipulation actions, the type of gripper and the type of robot.

C. Queries

Queries allow robots to investigate the knowledge base and retrieve existing data. A robot able to query the database has the capability to efficiently and intelligently perform tasks. In our case, a Python interface lets ROS users query the Semantic Knowledge Base using SPARQL[7].

D. The interlinking algorithm

In order to populate the knowledge base as much as possible, we interlinked instances of the proposed knowledge base with the ones of the Household Objects Database [12] provided by Willow Garage to the ROS community. Willow Garage created this database to provide, for each 3D surface model of the objects in the database, a large number of grasp points that are specific to a PR2 robot and its gripper.

In the considered sets, there can be instances that describe the same resource in the world. By interlinking these two instances, the two sets can be treated as one.

Interlinking can be done manually, if there are not many instances being created. Otherwise, an algorithm should be applied to automate the interlinking process. In [22] the interlink pattern of two data sets is found out applying two machine learning methods, the K-medoids and the Version Space. Although interlinking algorithms require interactions with users for the sake of the interlinking precision, computations of comparing instances are largely reduced than manually interlinking.

Algorithm 1 aims to interlink instances across two data sets *D* and *D'*. The algorithm first computes property/relation correspondences across two data sets (line 5). Then, instances property values are compared by referring to the correspondences (line 10). A similarity value *v* is generated upon all similarities of property values (line 11). If such a similarity is equal to or larger than a predefined threshold *T*, the two compared instances can be used to build a link with the relation *owl:sameAs* (line 12-14).

Algorithm 1 Interlinking Instances across Data Sets

```
   Input: Two Data Sets
   Output: Links accross Data Sets
 1: The data set D, D'; /*two data sets to be interlinked*/
 2: Similarity threshold T
 3: for Each property/relation in the data set D do
 4:     for Each property/relation in the data set D' do
 5:         Match properties/relations that are corresponding to each other and store as
    the alignment A
 6:     end for
 7: end for
 8: for Each instance in the data set D do
 9:     for Each instance in the data set D' do
10:         Compare instances' property values according to the correspondences of the
    alignment A;
11:         Aggregate all similarities between property values as a similarity value v
12:         if v >= T then
13:             The two compared instances are interlinked with owl:sameAs.
14:         end if
15:     end for
16: end for
```

IV. Experiments and Results

In order to prove the improvement introduced by the Semantic Knowledge Base, we implemented a sense-model-act framework.

A. Sense

A vision sensor acquires world data. We selected a Microsoft Kinect. RGBD images of the environment are converted into 3D point clouds and segmented into individual graspable objects using the ROS Tabletop segmentation tool[8] developed by Marius Muja.

1) Shape Detector: Each segmented object is represented by a point cloud. We use raw clouds and the input coordinates to extract the object and compute its shape.

[3]Extensible Markup Language (XML): http://www.w3.org/XML/

[4]Resource Description Format (RDF): http://www.w3.org/RDF

[5]Web Ontology Language (OWL): http://www.w3.org/2001/sw/wiki/OWL

[6]Word Wide Web Consortium (W3C): http://www.w3c.com/

[7]Simple Protocol and RDF Query Language (SPARQL): http://www.w3.org/TR/sparql11-query

[8]Tabletop Object Detector: http://www.ros.org/wiki/tabletop_object_detector

2) Object Recognizer: The tool allows robots to recognize objects. The tool is based on the ROS Tabletop segmentation tool. From the point cloud of an object, the tool extracts its meshes. Matching new objects meshes with existing ones, objects are recognized. The tool adds knowledge to the data set but it is not essential to solve manipulation tasks.

B. Model

1) Semantic Knowledge Base: After the environment mapping, the robot accesses the Semantic Knowledge Base to find a match between the segmented objects and the objects saved in the database. If the match exceeds a certain threshold, then the object is assumed to be recognized (if its 3D model exists) or detected (if only the shape is known). If (at least) the shape, the assigned action, and the gripper joints configuration are retrieved from the base, then a plan is generated containing the kinematics information required for the system to pass from the initial to the goal configuration.

2) Action Generator: If no information about the object, action, and joints configurations is stored in the database, then new data are generated. The framework extends the MoveIt! Simple Grasps tool to generate all possible grasp/push/place poses. Given the desired safety distance, the generator aligns the hand with the object principal axes and tries to manipulate the object around its center of mass starting from either the top or from the side of the object. It generates a list of possible poses of a gripper relative to the object, stores data in the Semantic Knowledge Base, and shares them among the robotics community.

3) Motion Planner: The planner adopts MoveIt! and the Kinodynamic Planning by Interior-Exterior Cell Exploration (KPIECE) [33] planner from the Open Motion Planning Library (OMPL) [34] library to compute a collision free path for manipulating the object. It chooses, from the set of generated actions, the first behavior that is kinematically feasible and collision-free and generates the plan. The collision check is performed on the 3D collision map created from the Kinect point cloud and takes into account collisions with the environment, joint limits and self collisions of the robot's arm. Any object manipulated by the robot is included in the robot model, in order to avoid collisions between the object and the rest of the environment during transport.

C. Act

1) Robot Controller: It activates the simulated/real engines that drive the robot.

2) Effect Evaluator: If we reason about arbitrary shapes, collisions or detachments can be induced by pre-selecting the manipulation configuration. To overcome the problem, the *Action Generator* generates the gripper's joints configuration required to perform the task. The *Motion Planner* plans movements. The *Robot Controller* actives robot motors and moves the robotic system along the planned path. During the execution of the task, failures may occur. The *Effect Evaluator* uses the information acquired by sensors to compare the system final state with the expected one. In case of mismatch, a learning module starts a trial and error routine that corrects the joints configuration and generates a new configuration. The configuration that allows the task achievement overwrites the existing one in the Semantic Knowledge Base.

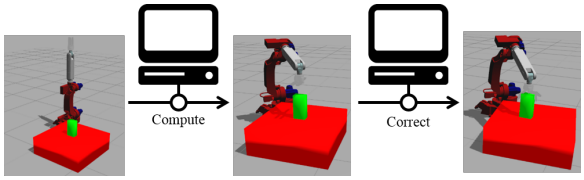


Fig. 2. The first experiment: the system compute the first pose and correct it until the achievement of the result

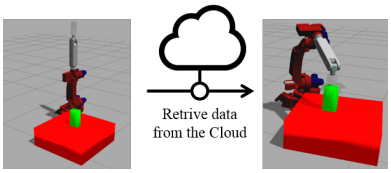


Fig. 3. The second experiment: the system retrive the information from the Cloud

D. Results

We validated the framework by performing experiments on a Comau Smart5Six robot equipped with a Schunk SDH gripper. Gazebo [35] was used as a simulated environment. In simulation, the robot has to grasp a parallelepiped.

During the first attempt, the system has no prior knowledge about the object. Manipulation data must be computed. The robot approaches the object and fails when trying to attempt the action. The trial-and-error approach allows the robot to manipulate the object (see Figure 2).

During the second attempt, the object is known and the Semantic Knowledge Base stores its manipulation data. The robot is able to manipulate the object on the first try (see Figure 3).

V. Conclusion and Future Work

As stated in [36], "Knowledge processing is an essential resource for autonomous robots that perform complex tasks in dynamic environments. Robots need advanced reasoning capabilities to infer the control decisions required for competently performing complex tasks like manipulation. Their knowledge processing system has to provide them with common-sense knowledge, with the ability to reason about observation of the environment, and with methods for learning and adapting over time". In this paper, from the study of humans actions when handling objects, an abstraction of the manipulation domain was formulated and encoded into an OWL ontology. A Semantic Knowledge Base was created to collect data representing the domain. We proved our approach by building a ROS framework able to associate manipulation actions to objects' shapes. Linking actions to shapes instead of objects' 3D CAD models increases the framework's reusability and guarantees its functioning when dealing with unknown objects. Tests were

performed in simulation and required the manipulation of I) a novel object II) the same object located in the Cloud.

As future work, the authors aim to extend the type of encoded actions storing not only translational pushes but also rotational ones. This implies the possibility to accomplish complex movements such as opening or closing doors. In fact, if a robot has to open a door, it will grasp the handle and perform a rotation. Moreover, while the *model* and *act* modules are well understood, we are actively working to fully define the *sense* module and to prove its well-functioning. We aim to improve also the *Effect evaluator* and to provide proofs of its well-functioning using other robots and grippers. The authors are performing tests in simple manipulation tasks on a real Comau Smart5Six robot.

REFERENCES

[1] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, pp. 1125–1134, 2010.

[2] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios: A Literature Review," Technische Universitt Dortmund (Fakultt Maschinenbau), Audi Stiftungslehrstuhl Supply Net Order Management, Tech. Rep., 2015.

[3] E. Tosello, S. Michieletto, A. Bisson, and E. Pagello, "A Learning from Demonstration Framework for Manipulation Tasks," in *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of.*, 2014, pp. 1–7.

[4] Z. Fan, E. Tosello, M. Palmia, and E. Pagello, "Applying Semantic Web Technologies to Multi-Robot Coordination," in *Workshop NRF-IAS-2014, Proceedings Of.*, Venice, Italy, 2014.

[5] G. Metta, G. Sandini, D. Vernon, D. Caldwell, N. Tsagarakis, R. Beira, J. Santos-Victor, A. Ijspeert, L. Righetti, G. Cappiello, G. Stellin, and F. Becchi, "The RobotCub project - an open framework for research in embodied cognition," *Humanoids Workshop, IEEE RAS International Conference on Humanoid Robots*, 2005.

[6] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 1349–1380, 2000.

[7] J. Pauli and G. Sommer, "Perceptual organization with image formation compatibilities," *Pattern Recognition Letters*, pp. 803–817, 2002.

[8] R. B. Rusu, Z. C. Marton, N. Blodow, A. Holzbach, and M. Beetz, "Model-based and Learned Semantic Object Labeling in 3D Point Cloud Maps of Kitchen Environments," in *Procceding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11-15 2009.

[9] D. Pangercic, M. Tenorth, B. Pitzer, and M. Beetz, "Semantic Objects Maps for Robotic Housework - Representation, Acquisition and Use," in *Procceding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[10] C. Goldfeder, M. Ciocarlie, and P. Allen, "The Columbia grasp dataset," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Kobe, Japan, 2009, pp. 1710–1716.

[11] A. Kasper, Z. Xue, and R. Dillman, "The KIT object models database: an object model database for object recognition, localization and manipulation in service robotics," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 8, pp. 927–934, 2012.

[12] M. Ciocarlie, K. Hsiao, E. Jones, S. Chitta, R. Rusu, and I. Sucan, "Towards reliable grasping and manipulation in household environements," in *in Proc. Int. Symp. Experimental Robotics*, Delhi, India, 2010, pp. 1–12.

[13] M. Tenorth and M. Beetz, "KnowRob: A Knowledge processing infrastructure for cognition-enabled robots," *International Journal of Robotics Research (IJRR)*, vol. 23, no. 5, pp. 566–590, 2013.

[14] M. Waibel, M. Beetz, J. Civera, R. D'Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft, "RoboEarth - A World Wide Web for Robots," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 2, pp. 69–82, 2011.

[15] H. Dang and P. K. Allen, "Learning grasp stability," in *in Proc. Int. Conf. Robot. Autom. (ICRA)*, Saint Paul, MN, 2012, pp. 2392–2397.

[16] J. Weisz and P. K. Allen, "Pose error robust grasping from contact wrench space metrics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, St. Paul, MN, USA, 2012, pp. 557–562.

[17] M. Popovic, G. Koostra, J. A. Jorgensen, D. Kragic, and N. Kruger, "Grasping unknown objects using an early cognitive vision system for general scene understanding," in *Proc. Int. Conf. Intell. Robot. Syst. (IROS)*, San Francisco, California, USA, 2011, pp. 987–994.

[18] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz, "The RoboEarth language: Representation and Exchanging Knwoledge about Actions, Objects, and Environments," in *23rd International Joint Conference on Artificial Intelligence (IJCAI) Special track on Best Papers in Sister Conferences (invited paper)*, 2013.

[19] T. Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001.

[20] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proc. Open-Source Software workshop of the International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[21] D. Di Marco, A. Koch, O. Zweigle, K. Haussermann, B. Schiessle, P. Levi, D. Galvez-Lopez, L. Riazuelo, J. Civera, J. M. M. Montiel, M. Tenorth, A. Perzylo, M. Waibel, and R. Van de Molengraft, "Creating and using RoboEarth object models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3549–3550.

[22] Z. Fan, "Concise Pattern Learning for RDF Data Sets Interlinking," Ph.D. dissertation, University of Grenoble, 2014.

[23] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on automation science and engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[24] L. Kaufman and P. Rousseeuw, *Statistical Data Analysis Based on the L1-Norm and Related Methods*, North-Holland, 1987, ch. Clustering by means of Medoids, pp. 405–416.

[25] V. Dubois and M. Quafafou, "Concept learning with approximation: Rough version spaces," in *Rough Sets and Current Trends in Computing: Proceedings of the Third International Conference (RSCTC)*, Malvern, PA, USA, 2002, pp. 239–246.

[26] L. Kunze, T. Roehm, and M. Beetz, "Towards semantic robot description languages," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shangai, China, 2011, pp. 9–13.

[27] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced Computer Vision With Microsoft Kinect Sensor: A Review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318 – 1334, 2013.

[28] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," in *Robotics and Automation, IEEE Transactions on*, vol. 5(3), 1989, p. 269279.

[29] T. R. Gruber, "A translation approach to portable ontology specifications," in *Knowledge Acquisition*, vol. 5(2), 1993, p. 199220.

[30] C. Schlenoff, E. Prestes, R. Madhavan, P. Goncalves, H. Li, S. Balakirsky, T. Kramer, and E. Miguelanez, "An IEEE standard Ontology for Robotics and Automation," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Vilamoura, Algarve, Portugal, 2012, pp. 1337 – 1342.

[31] C. I. Schlenoff, "An Overview of the IEEE Ontology for Robotics and Automation (ORA) Standardization Effort," in *Standardized Knowledge Representation and Ontologies for Robotics and Automation, Workshop on the 18th*, Chicago, Illinois, USA, 2014, pp. 1–2.

[32] E. Prestes, S. R. Fiorini, and J. Carbonera, "Core Ontology for Robotics and Automation," in *Standardized Knowledge Representation and Ontologies for Robotics and Automation, Workshop on the 18th*, Chicago, Illinois, USA, 2014, pp. 7–9.

[33] I. A. Sucan and L. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," in *Workshop on the Algorithmic Foundations of Robotics*, Guanajuato, Mexico, 2008.

[34] I. A. Sucan, M. Moll, and L. Kavraki, "OMPL Library," 2010. [Online]. Available: http://www.ros.org/wiki/ompl

[35] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Sendai, Japan, 2004, pp. 2149–2154.

[36] M. Tenorth, D. Jain, and M. Beetz, "Knowledge processing for cognitive robots," *KI - Knstliche Intelligenz*, vol. 24, no. 3, pp. 233–240, 2010.

The Effect of Synchronized Movement of the Tele-presence Robot on User's Perception

Hyemee Kang
Dept. of Industrial Design
Ewha Womans University
Seoul, Republic of Korea
mid_july@naver.com

Jee Yoon Lee
Dept. of Industrial Design
Ewha Womans University
Seoul, Republic of Korea
junedollri@naver.com

Min-Gyu Kim
R&D Division
Korea Institute of Robot and Convergence
Pohang, Republic of Korea
mingyukim@kiro.re.kr

Sonya S. Kwak
Dept. of Industrial Design
Ewha Womans University
Seoul, Republic of Korea
sonakwak@ewha.ac.kr

***Abstract*— Does synchronized movement of a telepresence robot with remote sender's movement improve telepresence communication between a sender and a receiver? For answering this question, we executed a 3 within-participants experiment for a telepresence robot with synchronized movement, a telepresence robot without movement and computer based video chat. The participants observed three different videos showing a remote sender and the perceived presence for each condition was measured in terms of telepresence, co-presence and social presence. The experimental results implied the importance of synchronization between robot and a remote sender in designing telepresence robot. The results showed that the participants felt more presence when interacting with the telepresence robot with synchronized movement than the robot without any movement or the computer-based video chat.**

Keywords— Human-Robot Interaction; Telepresence Robot; Synchronization; Co-presence; Telepresence; Social Presence

I. INTRODUCTION

It is importance to explore interaction design factors of a telepresence robot which has been developed for social communication between people at a distance [1], since a telepresence robot has multimodal communicative faculties that make it effectively deliver the presence of a remote sender to a receiver. Several researchers demonstrated that telepresence robot engages remote senders and receivers in robot mediated emotional interactions between remote senders and receivers. For instance, people experience comfort [2], social presence [3] and emotional empathy [4] in telecommunication through the telepresence robot.

The effectiveness of conveying the presence of a remote sender through the telepresence robot is significantly associated with physical embodiment of robot which other telecommunication devices such as smartphones and personal computers do not possess. The physical embodiment of the telepresence robot provides not only visual and auditory cues but also tactile information and physical motions that enhance presence.

This study focuses on an effect of telepresence robot movement synchronized with remote sender's movement. To do that, an experiment was designed in terms of three dimensions including telepresence, co-presence and social presence. Those have been recognized as major factors to be considered in many telecommunication systems.

In this paper, Section 2 discusses the preceding works on two issues: telepresence robot and synchronization. Section 3 introduces the study design including the hypotheses. In Section 4, the experimental results are shown in detail.

II. RELATED WORK

A. Telepresence Robot

Robots could be classified as a tele-operated robot and an autonomous robot according to the human intervention level [5]. Tele-operated robot is mainly focused on manipulating objects by a human operator in the remote site. Rather, telepresence robot is specialized with the purpose of the social communication between a remote sender and a receiver [6].

There are studies for the consequential effect of the human-robot interaction according to human intervention level by comparing between robot mediated human-human interaction and human-autonomous robot interaction. In simulated earthquake situation, people felt more comfortable toward a tele-operated robot than an autonomous robot [2]. People experienced more embarrassment and social presence when they had an interview through telepresence robots connected to interviewers than with autonomous robots [3]. Moreover, people had more emotional empathy to a tele-operated robot than an autonomous robot [4]. These studies showed that a telepresence robot enables people to emotionally interact with each other and effectively deliver the presence of the remote sender.

In telecommunication, presence is classified as three interrelated concepts, which can be described as 'you are there', 'it is here', and 'we are together'. Telepresence in its meaning 'you are there' characterizes the feeling that you are actually transported to a mediated world, the sense of being there inside the media. Co-presence in its meaning 'it is here' delineates the feeling that a remote sender comes to you while you are remaining, the sense of being connected to a remote sender [7]. Social presence in its meaning 'we are together' describes the feeling that a receiver and a remote sender shares emotion, the sense of being together with a remote sender emotionally [8]. For such concepts of presence, the previous studies in many

application domains such as persuasion [9] and education [10] have proved the effectiveness and usefulness of transmitting the presence of a remote sender by telepresence robot. In spite of that, investigating various interaction design factors is still required to augment the presence of a remote sender to a receiver for richer interactions between them.

B. Synchronization

Regarding presence in telecommunication, most of preceding studies have mainly focused on virtual environment through computers [11]. However, in the case of telepresence robot, it is essential to pay attention to the effect of physical embodiment on presence.

Sirkin and Ju [12] addressed this issue from point of view in synchronized movements between a remote sender and a robot. Their study revealed that synchronization between remote sender's movement and physical movement made by a telepresence robot significantly improved the receiver's interpretation for the remote sender's intention, compared to single-handed remote sender's movement or the physical movement made by a telepresence robot. Moreover, the physical movement positively influenced the perceptions of both the remote sender and the receiver. When the telepresence robot showed the receiver an unsynchronized movement, it led proxy-in-proxy problem that interrupted the receiver's interpretation about the remote sender's message. Ju et al. investigated the impact of synchronization focusing on a receiver's information interpretation in the telecommunication through the telepresence robot. Beside the information exchange, since a remote sender and a receiver are involved in the social interaction, further studies on emotional interactions through telepresence robot were needed.

III. Study Design

In this study, the experiment aims to verify how telepresence robot's synchronized movement can enhance telepresence, co-presence and social presence of a remote sender. The within-participants experiment was designed for telepresence robot with synchronized movement, telepresence robot without movement and a computer-based video chat. All participants were involved in three different conditions. The hypotheses are described as follows.

H1: A robot with synchronized movement of the remote sender will make the participants feel more telepresence than a robot without movement and a computer-based video chat.

H2: A robot with synchronized movement of the remote sender will make the participants feel more co-presence than a robot without movement and a computer-based video chat.

H3: A robot with synchronized movement of the remote sender will make the participants feel more social presence of the remote sender than a robot without movement and a computer-based video chat.

A. Participants

Eighteen participants (Male: 6 and female: 12) who have high technology acceptance were recruited. The participants were ranged in age from 22 to 30. They were educated at the college level on average.

B. Robot

In the experiment, FURo-iHome [13] was used as shown in Fig. 1. FURo-iHome is a cone shaped home service robot which was developed by FutureRobot Co., Ltd. It has a screen on a 1DOF neck which functions a tablet. A user can communicate with fellows via a IP camera, speakers and a microphone using the wireless/wired network.

Fig. 1. FURo-iHome

C. Procedure

We conducted video based experiment [14]. The participants were asked to watch three videos of two types of the telepresence robots and a computer-based video chat. The video stimulus contained video chat between two people through FURo-iHome. The conversation of each video was identical. During the conversation, the remote sender was nodding her head saying yes. In the case of synchronized movement, the robot nodded its screen synchronized of remote sender's nodding. In the other 2 cases did not move. Three videos were shown in a random order for counterbalance. After watching each video, a questionnaire about telepresence, co-presence and social presence was given.

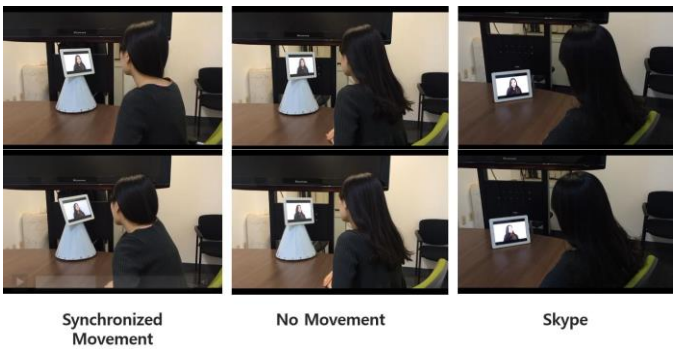


Fig. 2. Video stimulus

"This work was supported by Industrial Technology Innovation Program(Design Technology Innovation Program) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea)" (No. 10050008)

D. Measures

To measure the presence, telepresence, self-reported co-presence, perceived other's co-presence, social presence were used. The participants rated the robot on 29 different Likert-type items, which were drawn from Nowak et al. [15]. The items combined into 4 scales following reliability checks. We can report a Cronbach's Alpha of 0.97 for the telepresence, 0.92 for the self-reported co-presence, 0.97 for the perceived other's co-presence, and 0.96 for the social presence.

IV. RESULTS

A. Telepresence

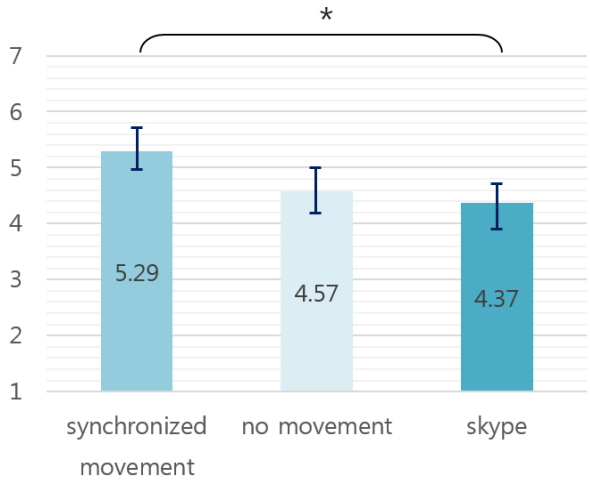


Fig. 3. Telepresence (Note: *, p<0.05)

As predicted by H1, a significant effect of synchronized movement on telepresence was found (F(2,34)=5.803, p<0.05). The robot with the synchronized movement of the remote sender (M=5.29, SD=.88) makes the participants feel more tele presence than the robot without movement (M=4.57, SD=1.01) or the computer-based video chat (M=4.37, SD=1.27).

B. Co-presence

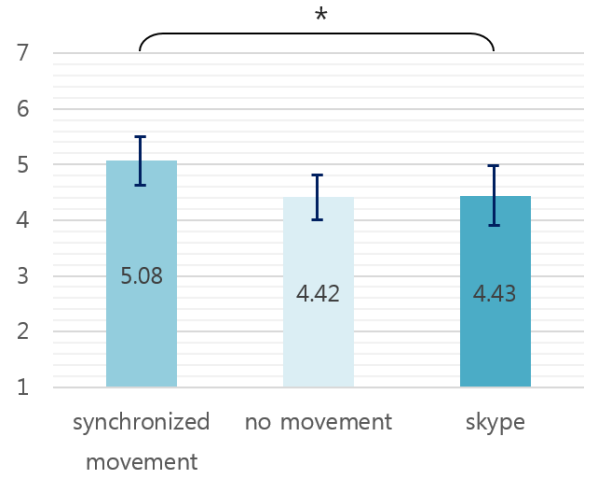


Fig. 4. Self-reported Co-presence (Note: *, p<0.05)

H2 was supported. A significant effect of synchronized movement on self-reported co-presence was found (F(2,34)=4.351, p<0.05). The robot with synchronized movement of the remote sender (M=5.08, SD=1.07) makes the participants feel more self-reported co-presence than the robot without movement (M=4.42, SD=.87) or the computer-based video chat (M=4.43, SD=1.31).

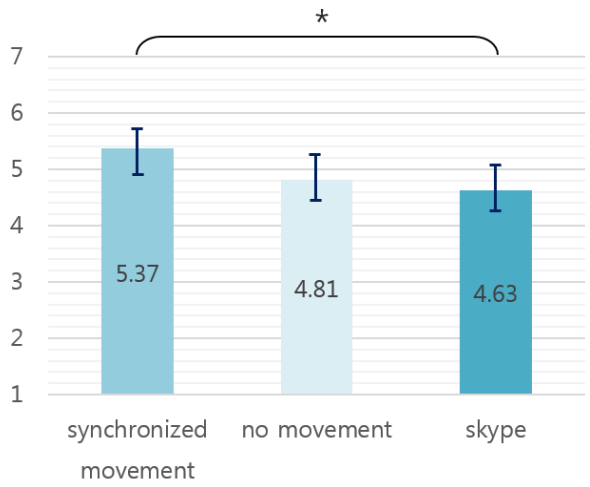


Fig. 5. Perceived Other's Co-presence (Note: *, p<0.05)

A significant effect of synchronized movement on perceived other's co-presence was found (F(2,34)=4.351, p<0.05). The robot with synchronized movement of the remote sender (M=5.37, SD=1.00) makes the participants feel more perceived other's co-presence than the robot without movement (M=4.81, SD=1.09) or the computer-based video chat (M=4.63, SD=1.11).

C. Social presence

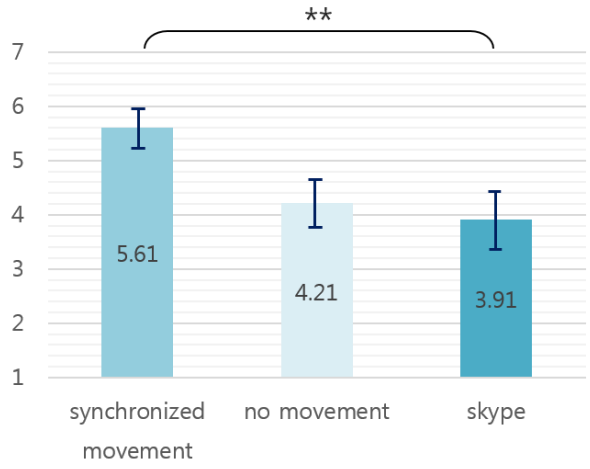


Fig. 6. Social presence (Note: **, p<0.01)

H3 was also supported. A significant effect of synchronized movement on social presence was found (F(2,34)=8.667, p<0.01). The robot with synchronized movement of the remote

sender (*M*=5.61, *SD*=.87) makes the participants feel more social presence than the robot without movement (*M*=4.21, *SD*=1.07) or the computer-based video chat (*M*=3.91, *SD*=1.25).

V. CONCLUSION

In this study, the effect of the telepresence robot with synchronized movement of a remote sender was examined. The participants felt more presence toward the telepresence robot with synchronization of a remote sender's movement than the telepresence robot without movement or the computer-based video chat. Through a remote sender's movement delivery, even a robot could impress people being in the same space. Although there are some limitations in this study, the results suggest that robot designers and engineers use the synchronized movement of a telepresence robot to effectively raise realism of the sender at a remote place. As this study was limited to the video-based short term study, follow-up experiments could be conducted in the live-based long term study. In addition, various range of ages and different cultural backgrounds will be involved in the follow-up study.

REFERENCES

[1] H. Lee, J. J. Choi, and S. S. Kwak, "The Impact of Telepresence Robot Types on the Perceived Presence of a Remote Sender," 2015.

[2] L. D. Dole, D. M. Sirkin, R. M. Currano, R. R. Murphy, and C. I. Nass, "Where to look and who to be: Designing attention and identity for search-and-rescue robots," In Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction, . IEEE Press, 2013 pp. 119-120

[3] J. J. Choi, Y. Kim and S. S. Kwak, "Are you embarrassed?: The impact of robot types on emotional engagement with a robot," In: Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, 2014.

[4] S. S. Kwak, Y. Kim, E. Kim, C. Shin, and K. Cho, "What makes people empathize with an emotional robot?: The impact of agency and physical embodiment on human empathy for a robot," In RO-MAN, 2013, pp. 180-185.

[5] H. A. Yanco and J. L. Drury, "A taxonomy for human-robot interaction," In Proc. AAAI Fall Symposium on Human-Robot Interaction, 2002, pp. 111-119.

[6] K. M. Tsui, M. A. Desai, H. Yanco, H. Cramer, and N. Kemper, "Measuring attitudes towards telepresence robots," International journal of intelligent Control and Systems, 16, 2011.

[7] T. B. Sheridan, "Descartes, Heidegger, Gibson, and God: toward an eclectic ontology of presence," Presence: Teleoperators and virtual environments, 8(5), 1999, pp. 551-559.

[8] K. Nowak and F. Biocca. "Defining and differentiating copresence, social presence and presence as transportation," Paper presented at the Presence 2001 Conference, Philadelphia, PA., 2001.

[9] H. Lee, J. J. Choi, and S. S. Kwak, "Will you follow the robot's advice?: the impact of robot types and task types on people's perception of a robot," In Proceedings of the second international conference on Human-agent interaction , 2014, pp. 137-140

[10] F. Tanaka, T. Takahashi, S. Matsuzoe, N. Tazawa, and M. Morita, "Telepresence robot helps children in communicating with teachers who speak a different language," In Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction, 2014, pp. 399-406

[11] R. Schroeder, "Social interaction in virtual environments: Key issues, common themes, and a framework for research," The social life of avatars: Presence and interaction in shared virtual environments. London: Springer-Verlag, 2002.

[12] D. Sirkin and W. Ju. "Consistency in physical and on-screen action improves perceptions of telepresence robots," Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction (2012), , Boston, USA, March 5–8, 2012

[13] FURo-iHome, FutureRobot Co., http://www.myfuro.com/furo-i/service-feature/. Accessed 1st. October 2015.

[14] S. N. Woods, M. L. Walters, K. L. Koay, and K. Dautenhahn, "Methodological issues in HRI: A comparison of live and video-based methods in robot to human approach direction trials," In Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on , 2006, pp. 51-58

[15] K. Nowak and F. Biocca, "The effect of the agency and anthropomorphism on users' sense of telepresence, copresence, and social presence in virtual environments," Presence 12(5), 2003, pp. 481-494

Reasoning on Communication Between Agents in a Human-Robot Rescue Team

Asil Kaan Bozcuoğlu[1], Fereshta Yazdani[1], Daniel Beßler[1], Bogdan Togorean[2] and Michael Beetz[1]
{asil, yazdani, danielb, beetz}@cs.uni-bremen.de
bogdan.togorean@student.utcluj.ro

***Abstract*— Humans and robots start to team up in rescue missions to overcome together the challenges arising in kinematics and locomotions by humans. In order to enhance the success of these heterogeneous teams, human operators should know how their robotic partners will behave under different conditions.**

In this paper we integrate the high-level plans of robots in a rescue team into the OPENEASE web application in order to reason about actions and behaviors of agents at different timepoints and different locations. By using already existing Prolog queries or the new ones that they create, people can ask questions such as *why*, *how* and *when* a robot has done a certain behavior. This kind of queries will be useful for operators to diagnose and to understand the behaviors of their partners. We show two different exemplary use cases in a human-robot team: In the first one, the robotic agent misinterprets the command and goes somewhere else. In the second one, it interprets the command correctly and is able to successfully reach the region-of-interest. By reasoning on these two cases, one can conclude which kind of commands can be misinterpreted by the robot.

I. INTRODUCTION

Robots are taking part in helping their human partners in critical missons such as extinguishing fires in forests and rescuing humans from dangerous situations. Here, the interests are going towards real world scenarios in which robots have to perform complex tasks together in the team. An example of such a real world scenario is presented in the project *SHERPA* [1]. This project aims at the interaction of mixed human-robot rescue teams in a hostile terrain where the human team leader has to interact with her robotic team in order to find injured persons. In such cases, the communication between agents should be as clear as possible in order to avoid from fatal casualties.

One way to accomplish a smooth communication, is that humans have to be more *preemptive*. In other words, humans should anticipate how robots will react to their commands in different circumstances. In this sense, investigating and diagnozing how robots will behave under different conditions can help for such kind of anticipation.

On the other hand, robots are still not very easy to access and play with for many people due to the factors such as expensiveness and safety reasons. In addition, the robot simulations are hard to setup and use for them because of the lack of programming skills and their complexity to install and use.

With the emergence of cloud-robotics applications such as [2][3][4], the cutting-edge robot plan/knowledge frameworks can also be reached over the web without a necessity to the installation. In OPENEASE [2], people can make Prolog queries in order to reason about kitchen experiments that the robots have done previously. Even people without any Prolog knowledge can choose different experiments and reason by using predefined Prolog queries that are inserted by developers.

In this paper, we investigate the collaboration between a human team leader and a quadcopter and the achievement of tasks in the heterogeneous team. A key objective in this paper is to reconstruct and comprehend the task execution based on the behaviors of the different agents. In order to achieve this goal we propose an approach of reasoning about robot activity descriptions in a cloud-based knowledge service with a heterogeneous team in a rescue application. The contributions of this paper as follows:

- we introduce previously presented systems based on the interaction of human-robot teams and on knowledge processing and algorithms for machine learning;
- we introduce a simulation-based rescue mission with a human team leader and a quadcopter in which we show exemplarily the exchange and process of information between the teammates;
- we introduce different experiment types and their results which will be added with a new set of Prolog queries, into OPENEASE;
- finally we show how these Prolog queries can be used by the human rescue team members to reason about their robotic partners behaviors in the past operations and simulations even without any Prolog knowledge.

II. RELATED WORKS

In real world scenarios, the demand of Human-Robot Interaction (HRI) is extremely high-scheduled. Working together as partners, exchanging information and assisting one to another to achieve common goals are key issues that must be addressed. One of the challenges is to provide human and robots with models of each other [6]. In recent years, many work have been focused on developing robots that work and interact directly with humans, as assistants or

[1]A. Bozcuoğlu, F. Yazdani, D. Beßler and M. Beetz are with the Institute for Artificial Intelligence, Universität Bremen, 28359 Bremen, Germany

[2]B. Togorean is with the Faculty of Automation and Computer Science, Technical University of Cluj-Napoca, 400604, Cluj, Romania

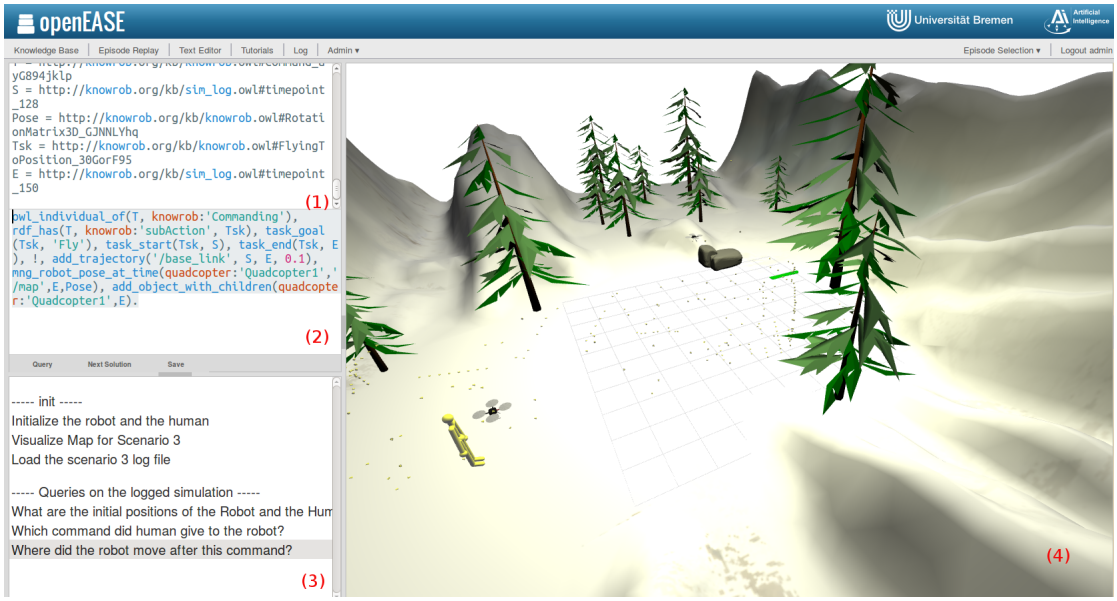


Fig. 2: The interface of OPENEASE. *The section (1)* is the Prolog console that one can see the previous queries and their results in the text form. *The section (2)* is the textbox that users can write new Prolog queries and execute them using *Query* button. In *the section (3)*, the predefined queries are listed. Users can query on these by clicking. In *the section (4)*, there is a 3D visual canvas. This canvas is updated when the users execute a Prolog query with a visual result.

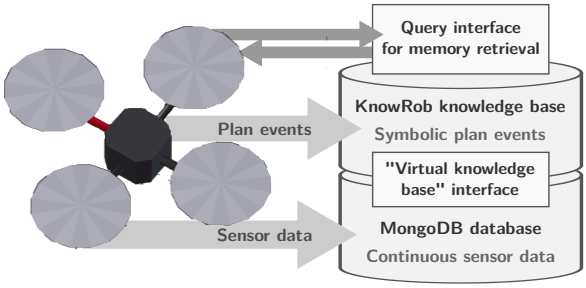


Fig. 1: The OPENEASE logging system. Image is based on [5].

teammates [7] [8] [9]. The field of Human-Robot Interaction research is addressed with communication, joint actions and human-aware execution that are challenging components and required by a smooth flow in human-agent activity [10]. In order to successfully accomplish common tasks robots require substantial knowledge about the environment they operate in and the objects they interact with. Such a knowledge system is offered in [11] that describes different kinds of knowledge and knowledge processing methods integrated in the system. A similar approach is done in [12] that works with the coordination of multiple robots in order to perform complex tasks assigned by people.

In the scope of robotics, there are some recent studies which put a special emphasis on knowledge processing. Saxena et al. [13], [14], introduce a learning methodology using natural languages in order to tell the robots how to accomplish a task. Moreover, there are some studies on using the world wide web as a deep knowledge source for robots for different goals such as concept learning [3] and task instructions acquisition [15].

Outside the scope of the robotics, Janowics et al. [16] propose a framework that combines machine learning algorithms with semantic web technologies. Wielemaker et al. [17] introduce a SWI Prolog-based web application similar to OPENEASE for the semantic web ontologies. They also introduce a SQL-like programming language, *SPARQL*, for researchers without Prolog knowledge.

In this paper we introduce a new approach, how robots' behavior can be better explored in order to enable a better communication in mixed human-robot teams. We investigate what challenges do arise when robots execute tasks in rescue missions and how these can be improved.

III. PROPOSED SYSTEM

In the proposed system, the quadcopter uses the Cognitive Robot Abstract Machine (CRAM) [18] framework for planning that enables developers to define and to execute cognition-enabled plans on robots. Testing such a system in real life is difficult, because it is time consuming and associated with high costs. Therefore at the moment we are using the simulation as a development tool for simulating the team in a visual physical environment. We are using Gazebo [19] which is a multi-robot simulator for in- and outdoor environments. For creating the logs of experiments, we use the same logging mechanism described in [5]. In this mechanism, symbolic-level knowledge such as the tree of tasks inside the high-level plan, task parameters and failure and success states of each goal are recorded into the Web Ontology Language (OWL) format (Figure 1). This format is a knowledge representation language for ontologies that describes taxonomies and classification networks and is defining a whole structure of knowledge for various domains.

The low-level sensory data which includes the necessary links to high-level tasks are stored into files. These files use the data-interchange format JavaScript Object Notations (JSON) which is easy for humans to read and write and for machines to parse and to generate.

After the execution of tasks in Gazebo we start logging the experiments. These logs are directly integrated into OPENEASE whose Prolog engine KNOWROB [20] is fully-compatible with the used logging scheme. First, we put high-level logs onto the FTP server as experimental data which uses OPENEASE. Second, we import the JSON files into mongoDB instance of OPENEASE. Optionally, it is also possible to manually add some predefined queries into the query library.

In the end, by logging in OPENEASE web interface, users can select *Rescue Operations* experiment logs and query about details of them either using predefined queries or entering their own queries into the Prolog console. In Figure 2 is an illustration of the OPENEASE web interface indicated which visualizes the activities and the world state during the task execution at specific timepoints. An example of a query which is formulated in a high-level description in order to be comprehensible for humans can look like "where did the robot move after a specific command". The corresponding Prolog query would look like (Figure 2)

```
?- owl_individual_of(T, kr:'Commanding'),
   rdf_has(T, kr:'subAction', Tsk), !,
   task_goal(Tsk,'Fly'),
   task_start(Tsk, S),
   task_end(Tsk,E),!,
   add_trajectory('base_link',S,E,0.1),
   mng_robot_pose_at_time
   (quadcopter:'Quadcopter1','/map',E,Pose),
   add_object_with_children
   (quadcopter:'Quadcopter1', E).
```

An explicit description of the query is given in the next section. Additionally, it is also possible to add new predefined queries by external users if administrative rights are available.

IV. EXPERIMENTAL SETUP

In a typical SHERPA rescue scenario, there usually exists a rescue team consists of many agents and a human. In this paper we focus on a team consisting of a quadcopter and a human simulated in Gazebo in order to give a basic idea of the possibilities of this system. The communication between the team members is through commands given by the human operator. By commanding, the human operator points to an area that the quadcopter should inspect. Whenever the quadcopter finds an injured person in this area, it calls the human partner for help.

In order to show how useful these experiments can be used inside OPENEASE, we show two different human-robot communication scenarios in two different landscapes. In both scenarios, the human commands the robot to navigate to a particular area with different natural language instructions.

V. USE CASES

All of the queries explained in this section are also predefined in the corresponding experiment in OPENEASE

Fig. 3: The start positions of the agents in Figure V-A.

with the natural language descriptions.

A. A Misunderstanding Case

In this case, a robot and a human are exploring a valley in the middle of mountains. In order to see the start position of these agents in order to be able to understand their behavior step by step, we can make a Prolog query as follows:

```
?- owl_individual_of(Exp, kr:'Experiment'),
   rdf_has(Exp, kr:'startTime', ST), !,
   add_stickman_visualization(xs:'Human1', ST),
   mng_robot_pose_at_time(q:'Q1','map',ST,P),
   add_object(q:'Q1',ST).
```

Afterwards, the positions of each agent is visualized and can be seen in the canvas (Figure 3).

To have a look into the command which the human gave, we execute the following query:

```
?- owl_individual_of(T, knowrob:'Commanding'),
   rdf_has(T, knowrob:'taskContext', Goal).
Goal = Navigate the area behind me
```

Finally, in order to see how the robot reacts and proceeds after getting the command "Navigate the area behind me", we will query the subtask of this corresponding *Commanding* task with the context *Fly*, then, we will look for the position of the robot at the end of this subtask:

```
?- owl_individual_of(T, knowrob:'Commanding'),
   subtask(T, Tsk), task_goal(Tsk, 'Fly'),
   task_start(Tsk, S), task_end(Tsk, E), !,
   add_trajectory('/base_link', S, E, 0.5),
   mng_robot_pose_at_time(q:'Q1','/map',E,P),
   add_object(q:'Q1',E).
```

As seen the final position of the robot in Figure 4, the robot has misinterpreted the command and gone in front of the human partner.

This example shows a misinterpretation and miscommunication between agents which can have fatal effects for the team and for the whole mission.

B. A Successful Communication Between Agents

In the second use case, again, a human team member and a robot are trying to find a victim in a valley passed by a river. When we make the same query for the agents' initial positions which we have also used in the first use case, we

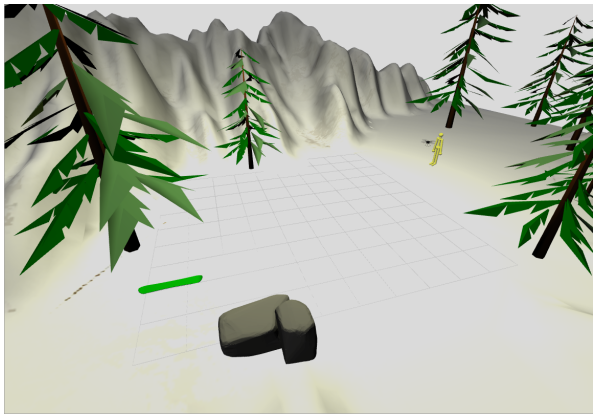


Fig. 4: The end positions of the agents in Figure V-A.

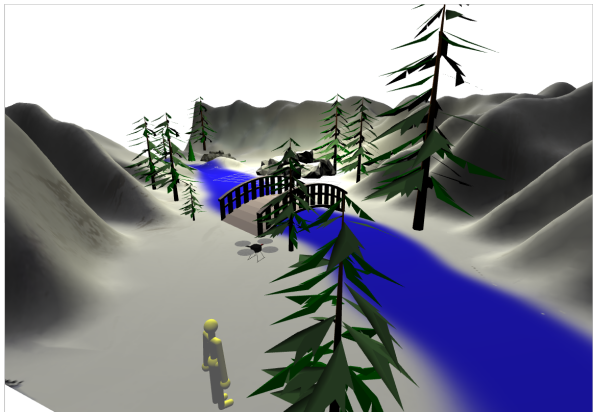


Fig. 5: The start positions of the agents in Figure V-B.

can see that the agents are standing close nearby the bridge (Figure 5).

One more time, for the command that the human gives we make a query with:

```
?- owl_individual_of(T, knowrob:'Commanding'),
   rdf_has(T, knowrob:'taskContext', Goal).
Goal = Explore 200 mt far away
```

If we look at the final position of the robot after using the command "Explore 200 mt far away", this time, we can see that the robot has successfully interpreted the command and gone to the region-of-interest (Figure 6).

In addition to the use cases, one essential result that a human team member can derive from these cases is, that a robot can successfully reach the region-of-interest when the given command includes an absolute position such as "Navigate the area that is 500 mt ahead". But if it includes some relative position definitions according to the team leader or to the robot itself, it is highly possible that the robot fails to accomplish the given command.

VI. Conclusion

In this paper, we have proposed a new experiment type to OPENEASE web application. The experiment was based on a collaboration with a human-robot team in a rescue scene in which the human member instructed the robot to look for injured persons in a scene. By using this kind of experiments, users, even without a technical background, can analyze, diagnose and debug behaviors of robots when they are commanded. In future, we are planning to extend the number of these rescue experiments with different scenarios so that humans can reason about the behaviors in an extended dataset in order to have a better anticipation and comprehension of the robot behaviors.

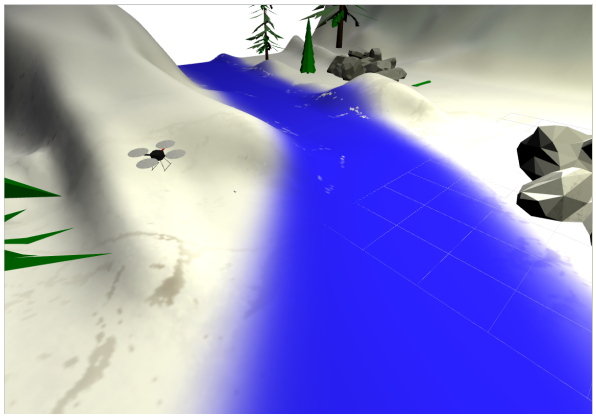


Fig. 6: The end position of the robot in Figure V-B.

Acknowledgements

This work is supported by the European Commission's FP7 project, *SHERPA*, with grant number 600958.

References

[1] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic†, R. Siegwart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty, A. Kleiner, V. Lippiello, A. Finzi, B. Siciliano, A. Sala, and N. Tomatis, "The sherpa project: smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, College Station, Texas, USA, Nov. 5-8 2012.

[2] M. Beetz, M. Tenorth, and J. Winkler, "Open-EASE – a knowledge processing service for robots and robotics/ai researchers," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, Washington, USA, 2015, finalist for the Best Conference Paper Award and Best Cognitive Robotics Paper Award.

[3] O. S. A. J. D. K. M. Ashutosh Saxena, Ashesh Jain and H. S. Koppula, "Robo brain: Large-scale knowledge engine for robots," arXiv, Tech. Rep., 2014, http://arxiv.org/abs/1412.0691.

[4] L. Riazuelo, M. Tenorth, D. D. Marco, M. Salas, L. Mösenlechner, L. Kunze, M. Beetz, J. D. Tardos, L. Montano, and J. M. M. Montiel, "Roboearth web-enabled and knowledge-based active perception," in *IROS 2013 Workshop on AI-based Robotics*, Tokyo, Japan, November 7th 2013.

[5] J. Winkler, M. Tenorth, A. K. Bozcuoglu, and M. Beetz, "CRAMm – memories for robots performing everyday manipulation activities," *Advances in Cognitive Systems*, vol. 3, pp. 47–66, 2014.

[6] T. B. Sheridan, "Eight ultimate challenges of human-robot communication," in *Robot and Human Communication, 1997. RO-MAN'97. Proceedings., 6th IEEE International Workshop on*. IEEE, 1997, pp. 9–14.

[7] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte, and S. Thrun, "Towards personal service robots for the elderly," in *Carnegie Mellon University*, 2000.

[8] T. Längle, T. Höniger, and L. Zhu, "Cooperation in human-robot-teams," 1997.

[9] I. R. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto, "An affective mobile robot educator with a full-time job," 1999.

[10] G. Klein, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich, "Ten challenges for making automation a "team player" in joint human-agent activity," *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 91–95, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1109/MIS.2004.74

[11] M. Tenorth and M. Beetz, "KnowRob – Knowledge Processing for Autonomous Personal Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4261–4266.

[12] Y. Mori, Y. Ogawa, A. Hikawa, and T. Yamaguchi, "Multi-robot coordination based on ontologies and semantic web service," in *Knowledge Management and Acquisition for Smart Systems and Services*, ser. Lecture Notes in Computer Science, Y. Kim, B. Kang, and D. Richards, Eds. Springer International Publishing, 2014, vol. 8863, pp. 150–164. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-13332-4_13

[13] Dipendra K Misra, Jaeyong Sung, Kevin Lee, Ashutosh Saxena, "Tell Me Dave: Context-Sensitive Grounding of Natural Language to Mobile Manipulation Instructions," in *Robotics: Science and Systems (RSS)*. IEEE, 2014.

[14] Jaeyong Sung, Bart Selman, Ashutosh Saxena, "Synthesizing Manipulation Sequences for Under-Specified Tasks using Unrolled Markov Random Fields," in *International Conference on Intelligent Robotics and Systems (IROS)*. IEEE, 2014.

[15] M. Tenorth, D. Nyga, and M. Beetz, "Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web," in *IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, May 3–8 2010, pp. 1486–1491.

[16] K. Janowicz, F. van Harmelen, J. A. Hendler, and P. Hitzler, "Why the data train needs semantic rails," *AI Magazine*, 2014.

[17] J. Wielemaker, W. Beek, M. Hildebrand, and J. van Ossenbruggen, "Cliopatria: A logical programming infrastructure for the semantic web," *Semantic Web Journal*, 2015, under review.

[18] M. Beetz, L. Mösenlechner, M. Tenorth, and T. Rühr, "Cram – a cognitive robot abstract machine," in *5th International Conference on Cognitive Systems (CogSys 2012)*, 2012.

[19] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, pp. 2149–2154 vol.3, 2004.

[20] M. Tenorth and M. Beetz, "KnowRob – A Knowledge Processing Infrastructure for Cognition-enabled Robots," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566 – 590, April 2013.

Exploration of Human Reactions to a Humanoid Robot in Public STEM Education

Lixiao Huang and Douglas Gillan
Department of Psychology
North Carolina State University
Raleigh, NC, USA
lhuang11@ncsu.edu

Daniel McDonald
Humanoid Robot Project
IEEE ENCS Robotics and Automation Society
Raleigh, NC, USA
daniel.w.mcdonald@gmail.com

***Abstract*— Humanoid robots have become a hot topic for robot design in the service and entertainment industry. However, there is a gap between humans' rich virtual exposure to humanoid robots through the media and their actual interaction experiences with them. To provide research support for humanoid robot design, the present paper explored the behavioral pattern of humans, dialog themes, and emotional responses in interaction with a humanoid robot that is capable of face recognition and conversations at two public settings: a park (50+ people) and a charter school (about 360 people). Results showed that major interaction activities of the adult dominant group at the park included looking at the robot, talking to the robot, talking to others about the robot, and taking photos. Children at the school did similar activities except taking photos, and they showed strong desire to interact with the robot and rich emotional responses. Major dialog input themes from the participants included greeting, asking about the robot's identity (e.g., age, origin), testing the robot's knowledge and capabilities, talking about preferences and opinions, and correcting the robot's conversation errors. Observed emotional responses included liking, surprise, excitement, fright, frustration, and awkwardness. Overall, the children showed more positive emotions than negative emotions. The study provided evidence that adults and children interact with the humanoid robot the way they interact with other humans, and it provided evidence supporting the uncanny valley effect. Future research will explore more populations and seek more rigorous research methods.**

Keywords— humanoid robot; children; public STEM education; human-robot interaction

I. INTRODUCTION

Humanoid robots have become a hot topic for robot design in the service and entertainment industry. Considering this trend, it is likely that today's children will be a group of people who use robots regularly in the next 20-30 years. Their interest and exposure to robots will have a great impact on the robot industry. This trend is further evidenced by the US Government's inclusion of robotics in its efforts to promote STEM education in the United States. People have had exposure to humanoid robots since the 20th century through movies (e.g., Metropolis in 1927; Bicentennial Man in 1999; The Stepford Wives in 2004; Ex-Machina in 2015), TV shows (e.g., Small Wonder in the 1980s; Humans in 2015), and YouTube videos (e.g., Geminoid DK & Ishiguro). However, people rarely have experiences of interacting with a humanoid robot face-to-face. There is a great need for research on how humans interact with humanoid robots to support the practice of designing humanoid robots.

The current humanoid robot project provides opportunities to educate adults and children about science and engineering through actual interaction with a humanoid robot. This research provides evidence to answer the following research question: How do humans behaviorally, verbally, and emotionally interact with a humanoid robot in public environments?

II. LITERATURE REVIEW

A. Human Computer Interaction and Anthropomorphism

Reeves and Nass found that humans tend to interact with their computers the way they interact with other humans [1]. Therefore, it is reasonable to predict that humans will interact with a humanoid robot in a similar way to how they interact with other humans.

Espley, Waytz and Cacioppo [2] proposed a three-factor theory that can be used to predict when people are likely to anthropomorphize a robot and when they are not: (a) The availability of knowledge of anthropomorphism, (b) the motivation to make sense of the behaviors of a robot, and (c) the need for social connection. These three factors are not difficult to find in human interactions with a humanoid robot. First, the human-like appearance of a humanoid robot boldly suggests to a human viewer, in a way that a non-humanoid robot may not, that the robot will behave in a human-like manner. Second, humans have a natural tendency to try to make sense of the world. Even though many people today have few experiences interacting with humanoid robots, interacting with other humans is an everyday occurrence. It is natural to transfer knowledge of interactions with other humans to interactions with a robot made to resemble a human. This forms a backdrop in which to assimilate and accommodate new, nonhuman, robot behaviors into the existing knowledge system of human-human interaction experiences. Third, according to the self-determination theory [3], humans have an inborn nature to connect with social contacts. If this is the case, it is not difficult for humans to anthropomorphize a humanoid robot to make this connection. Therefore, it is hypothesized that humans will, by default, interact with a humanoid robot like interacting with another human. This may include showing a range of emotional reactions such as pleasure at meeting a

new acquaintance, confusion and awkwardness at lack of understanding or communication failure, and antagonism, if social relationships are strained, for example, by bragging.

B. *Children and Robots Interaction*

There has been some research on human interaction with a humanoid robot [4]. Related research [5] on using robots in autism research mainly focuses on behaviors that increase and maintain children's engagement in interacting with the robots, such as eye-to-eye gaze. Engagement is not only an issue for autistic children, but for public interest in STEM education.

Investigating the reactions of normal children and adults to a robot dog at a shopping mall [6] showed that children developed positive emotions toward the robot dog at the visceral level, at the behavioral level, and at the reflective level. The children became excited when they first saw the robot dog, then they played with the robot dog, and they expressed the wish to bring the dog home. However, a robot dog has much fewer potential functions than a humanoid robot, and plays a significantly different role from a humanoid robot. It is expected that children would react differently to a humanoid robot from a robot dog. It is hypothesized that children will also interact with a humanoid robot in a way that is similar to how they interact with other humans.

C. *Uncanny Valley*

The uncanny valley effect is a phenomenon whereby a human's liking of a robot increases as the robot's resemblance to a human increases, up to a point. When a robot (or other human likeness, such as a cartoon or painting) closely resembles a human yet differs in some barely noticeable way, the human's liking of the robot suddenly drops and is replaced by a feeling of extreme dislike [7]. The turning point may vary across populations, but it is likely to observe negative emotions toward a humanoid robot if its human likeness falls in this close, but imperfect, range.

The purpose of this research is to explore human-robot interaction in public settings through three aspects: (1) The human behavioral patterns, (2) human dialog text, and (3) human emotional responses to a humanoid robot during interaction.

III. METHOD

The current paper consists of two parts: an informal preliminary observation at a park picnic and a formal study at a charter school, both at an eastern city in the US. Without statistical measures, the observational data collected at the park were not intended for research and generalization, rather to develop a rubric to facilitate quantitative data collection for the second study at a charter school. However, the actual arrangement at the school turned out to be so different from the park that the researchers developed a new rubric.

A. *The Humanoid Robot "KEN"*

KEN is a humanoid robot made from a mannequin upper body and head, with built-in computers. He detects faces and learns to recognize the people he meets. He can carry on a conversation with a human. KEN can move his neck horizontally or vertically, which allows for face tracking and human-like head gestures. A picture of KEN is shown in Fig. 1 and more information can be found in this website: http://sites.ieee.org/encs-humanoid/

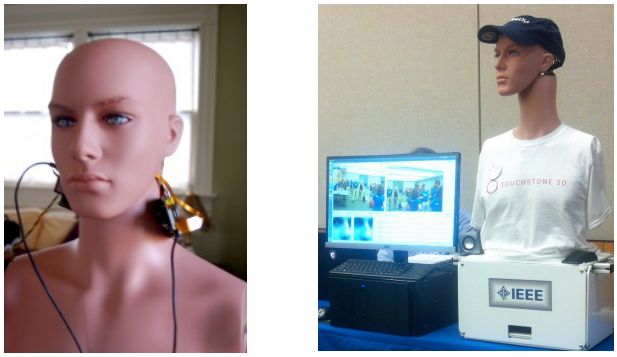


Fig. 1. Humanoid robot KEN (left photo by Kiko McDonald, under testing at a workplace; right photo by Lixiao Huang, normal setup for demo)

1) Vision. KEN has authentic blue eyes. KEN is constantly searching the video image frames from his eye cameras for human faces. He moves his head to center a face in his gaze. In the background, he records information about the faces he sees to allow him to recognize the face again and to associate the face with a name. The computer monitor shows the scenes KEN sees through his eye cameras, as well as identified faces.

2) Speech. KEN uses voice activity detection and speech recognition to record spoken phrases and translate them to text. The artificial intelligence system based on the ALICE chatbot processes the text into a response, which is spoken back using the eSpeak text to speech synthesizer. KEN has a speaker embedded in his chest. He hears through a microphone on the table or a cell phone receiver. When the background noise is low, the audience can speak directly standing in front of KEN. When there is a certain level of noise, the audience needs to pick up the microphone on the table to speak to KEN. Voice activity detection is done by a naïve sound intensity threshold algorithm. Manual voice activity detection is possible for acoustically challenging environments by using either a cell phone app or manually unmuting and muting the microphone. Speech to text conversion is performed by the Google Web Speech API. The computer monitor shows the transcribed text KEN receives from the audience and the machine generated responses that he speaks.

B. *Data Coding*

All codings used the data-driven method [8], in which the categories were created based on what was observed from the interaction. The three types of codings are:

1) Behavioral pattern coding. The major interaction activities the audience engaged in with the humanoid robot KEN were coded, or example, looking at the robot or talking to the robot.

2) Dialog text coding. When people speak to KEN, his voice activity detection and speech recognition translate speech into text. Then he generates a text response which he says to the audience using a synthesized voice. The transcribed dialog text is stored in KEN's computer. Only dialog text from humans was analyzed, using the verbal data analysis method [9], to get the dialog input themes of humans talking to a humanoid robot, for example, greeting, asking about name, hobby, origin, and language.

3) Emotional response coding. The emotional responses were coded from observation of the audience at the site as they interacted with the humanoid robot KEN, for example, liking, excitement, fright, and curiosity.

IV. Observation 1: At a Park Picnic

The first observation took place at a community outreach event of a local professional organization in summer 2015: a 4-hour picnic at a park. The participants were instructed to eat BBQ first and then come to interact with KEN. The food was constantly available and people were free to leave at any time.

A. Participants

About 50-70 people attended the event, including members of the sponsoring organization and their families and friends. Observation notes recorded 22 people's interactions with the robot. The participants included males and females; white, Asian, black or south African; estimated age from 3 to 70; children, high school students, college students, graduate students, young professionals, and senior professionals. Their social units included individuals, father and son, mother and son, father and daughter, a family of three or four, adults with their older parents and son, and friends.

B. Procedure

The humanoid robot KEN and the computer monitor were set up ahead of the event. When people approached the robot, a robot developer introduced KEN's abilities to see and converse, and he gave a quick demonstration of speaking to the robot KEN. Depending on the background noise level, the demonstration involved directly talking to KEN or talking through the microphone held in hand. Then the audience took over and interacted with the robot for different lengths and in various ways that naturally occurred. The observation started when people began approaching the robot.

C. Results

1) Behavioral pattern. The categories of the observed behavioral pattern are listed in Table I.

TABLE I. Behavioral Categories

Categories	*Notes*
Talking	• Talking to KEN directly to the face • Talking to KEN via the cell phone • Talking to KEN via the microphone held in a hand • Explaining to one's company what KEN is doing • Making comments about KEN when thinking aloud - e.g., when a man talked to KEN and KEN did not respond correctly, the man said to himself while looking at KEN, "He does not know how to respond." • Making comments about KEN to others • Discussing about KEN within a small group standing nearby • Asking the developer questions about KEN • Encouraging others to talk to KEN
Looking	• Looking at KEN waiting for response • Looking at the back of KEN • Looking at the inside of KEN • Looking at the computer monitor of what KEN sees • Looking around for the next interested person to pass on the microphone • Looking at other people (strangers, family, or friends) interacting with KEN and listening to their dialogs
Taking photos	• Taking photos of KEN • Taking photos with KEN

The length of direct engagement with KEN was normally less than 5 minutes. Communication with the developer and watching others interacting with KEN could reach 15 to 20 minutes. Many people approached KEN multiple times.

2) Dialog themes. The transcribed dialog text included 166 input records from the audience, as well as 78 times face recognition commands sent from the internal AI system. The dialog input themes included greeting, self-introduction, testing KEN's capabilities (e.g., math, telling a joke), asking about facts about KEN (e.g., name, age, preference of food, and opinions on politics), see Table II.

TABLE II. Dialog Themes

Categories	*Notes and Examples*
Greeting	• First contact - e.g., "Hello", "Hi" • Farewell - e.g., "See you again have to go."
Asking about KEN's identity	• Name - e.g., "What is your name?" • Age - e.g., "How old are you?" • Hobby - e.g., "What is your hobby?" • Origin - e.g., "Where are you from?" • Language - e.g., "Do you speak German?" • Experiences - e.g., "Have you been to the beach?" • Friend - e.g., "Who is your best friend?"
Self-introduction	• Self-introducing things they asked KEN about: name, hobby, preferences, etc. • "My name is [Maria]."; "I build robots."; "My favorite food is oatmeal."
Asking KEN's opinions	• "How do you feel about…?"; "Can you tell me a little bit about…?"; "What is life?" • E.g., Java, politics, life, etc.
Asking KEN's preference	• "Do you like…?"; "What kind of…do you like?"; "What is your favorite…?" • E.g., food, movie, pattern, and sport.
Testing Ken's knowledge and capability	• Recognizing color - e.g., "What color is my hat?" • Calculation of math - e.g. "What's two plus two?" • Telling a joke - e.g., "Can you tell a joke?" • Memory - e.g., "I have seen you before, do you remember me?" "Do you recognize me?" • Dreaming - e.g., "What do you dream about?"
Correcting KEN	• Correction included name, color, etc. • E.g., "No you've got me confused I'm [Tom]"; "No, it is beige."
Comments and Teasing	• Expressing emotion, e.g., "We love you." • Teasing - e.g., "You look a lot like a fellow named Ken."

Using the verbal data analysis method resulted in a frequency pattern of the dialog themes in Fig. 2. Greeting, self-introduction of name, talking about preferences and opinions, and testing KEN's capabilities were the most frequent dialog themes.

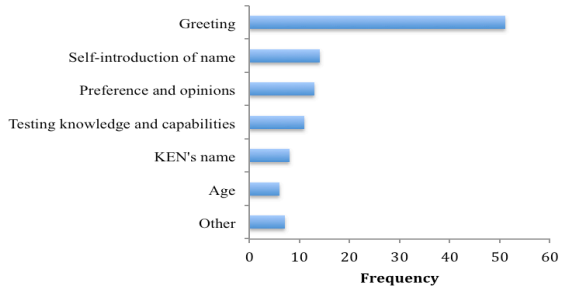


Fig. 2. Dialog themes from the audience at the park

3) Emotional responses. The observer noted a variety of emotions based on facial expression, behavior pattern and dialog text (see Table III).

TABLE III. EMOTIONAL RESPONSE CATEGORIES

Categories	*Notes and Examples*
General affect	• Interacting with the robot indicated liking • Their smiling facial expressions showed positive emotions.
Excitement	• Positive comments - e.g., "This is so interesting" • Risen eyebrows, opened mouth, cheerful smile
Curiosity	• Asking questions about the robot • Checking the inside of the robot
Intimidation	• Comment - e.g., "the robot looks creepy" • A child dared not to talk to the robot directly but asked his dads to talk to the robot.
Annoyance/ awkwardness	• When the robot failed to hear the input correctly, male adults showed annoyance, and female adults showed awkwardness.
Frustration	• When robot repeatedly failed to recognize faces correctly or say names correctly, some people showed frustration.

V. OBSERVATION 2 : AT A SCHOOL

The second observation was conducted at a charter school. The event took place from 10am to 2pm, divided into seven time slots. The teachers in each grade signed up for one time slot and brought the entire grade level to a large multipurpose room where KEN was located. Kindergarten through third grade signed up for 20-minute slots. Fourth through seventh grades signed up for 40-minute slots with a more in-depth presentation. The final presentation slot combined sixth and seventh grades. The demos for each group were arranged back-to-back with a five-minute transition between groups. The teachers of each group repeatedly instructed their students to be quiet throughout the event.

A. Participants

A total of 360 children from kindergarten to 7th grade participated in the event, along with 21 teachers (approximately three teachers for each grade level). The number of students for each grade level is listed in Table IV. The major race categories of the children included white and black or African American. Gender appeared to be distributed evenly.

TABLE IV. PARTICIPANTS' GRADE LEVEL, AGE, AND NUMBER

Time	*Grade Level*	*Age*	*Number*
10:00-10:20	First grade	6-7	50
10:25-10:45	Third grade	8-9	50
10:50-11:10	Kindergarten	5-6	40
11:15-11:35	Second grade	7-8	50
11:40-12:20	Fourth grade	9-10	55
12:25-13:05	Fifth grade	10-11	50
13:10-13:55	6th/7th grade	11-13	65

B. Measures

The data sources for this study included three sources: (1) Handwritten observation notes including questions asked and physical and emotional reactions to the robot; and (2) transcribed dialog input text - what KEN actually perceived and how he responded.

C. Procedure

The robotics team set up KEN at a table adjacent to the west wall and projected a computer screen to the east wall. At the beginning of each time slot (see Table IV), teachers brought in children and let them sit at the center of the room, facing the east wall. First, the robot team leader greeted the children and went through the following steps: (1) Asking a few questions, including what do engineers do and what do engineers build, (2) showing a 90 second YouTube video introducing what engineers do, (3) showing a YouTube video of a self-driving car that the robot team leader worked on before (this step was only for 4-7th graders), (4) directing attention to the humanoid robot KEN. Questions were accepted from the audience during these steps.

Second, the children were asked to turn to face KEN at the west wall and the robot developer went through the following steps: (1) Briefly introducing KEN, (2) demonstrating speaking to KEN, (3) using two American Girl dolls (Emily & Liberty) to demonstrate face recognition and ask the students the difference between a doll and a robot, (4) asking volunteers to come up to talk to KEN, one person at a time, and (5) answering more questions from the audience.

D. Results

1) Behavioral pattern. The behavorial pattern consisted of greeting, volunteering, and reacting to KEN's performance and the presenter's information about KEN (see Table V).

TABLE V. CHILDREN'S BEHAVIORAL PATTERNS DURING INTERACTION

Categories	*Notes and Examples*
Greeting	• Waving one hand or two hands to KEN when first met KEN, saying "Hello/Hi KEN"
Asking questions	• During the whole event, students consistently raised hands to ask questions. When not picked, they raised again. • At the end of the session, still more than half of the students raised hands to ask questions, but time only allowed a small number to speak up.

Categories	*Notes and Examples*
Interacting with KEN	• When they were asked to have someone volunteer to talk to KEN, everybody raised their hands up high. • A fourth grade student begged to let him try, saying, "Please, please, let me try it. I love robots!" • A second grade boy said, "I got this.", and stepped in front of KEN before he was called to come up. • When it was time to leave, each group had more than 10 students who got up and stood in front of KEN, either talking to KEN or just looking around KEN and the computer monitor until their teachers urged them to leave; some students waved hands at KEN and trying to get KEN's attention when they were lined up to leave.

2) Dialog themes. The dialog data consisted of two sources during the event: (1) Dialog text of the conversation with KEN stored on KEN's computer (see Fig. 3), including 155 input items; and (2) questions for the presenters (see Table VI).

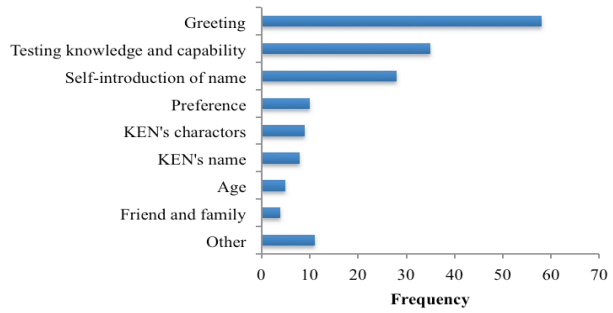


Fig. 3. Dialog themes from the audience at the charter school

TABLE VI. CHILDREN'S QUESTIONS ABOUT THE ROBOT BY GRADE

Questions	*K*	*1*	*2*	*3*	*4*	*5*	*6/7*
How long does it take to build/program a robot?	x			x			x
Why does he not have legs? Where are his arms? Why didn't you build legs?	x		x	x			
How did you make it [the robot]?	x						
Can he laugh at jokes? Does he know any jokes?		x					x
What is he made of?		x					
How can he move his head? How do you make the robot move?			x				
Would the robot become too hot if there is a computer running inside?			x				
Does he know math?			x				
How does he memorize things?			x				
How does he know what is the Internet?				x			
Is it a prototype? Are you going to make another one?				x			
What is the purpose of making it?				x		x	
How does it take pictures of us?					x		
If you program the robot, what programming language do you use?					x		
How did you come up with his name? What does KEN mean?						x	x
Is there anything on his chest? Can you lift up his shirt?						x	
How did you make the voice inside? Can you change his voice? How did you build him to talk?						x x	
What is a good way to start to be an engineer?						x	
If you ask silly questions, will he answer or get confused?							x
Is there an ear on KEN?							x
How far can KEN hear?							x
Is KEN aware of his gender?							x
Does KEN know about Siri?							x

3) Emotional response coding. Children showed strong emotional responses during the event (see Table VII). A few critical moments included: (1) Introducing KEN's camera embeded eyes, (2) showing KEN's math ability, (3) when KEN failed to respond appropriately, and (4) when KEN said something interesting. Emotional responses included emotion related behavors and emotional comments about KEN.

TABLE VII. EMOTIONAL RESPONSES OF THE STUDENTS AT THE SCHOOL

Categories	*Notes and Examples*
Liking	• Several fourth grade shouted, "I love robotics" • A third grade child said, "I will miss you KEN"
Excitement	• Evidences for excitement: (1) loud sound by clapping, laughing, and shouting, "Wow", "That's very cool!", and "Awesome!"; (2) opening mouths, widening eyes, raising eyebrows, and hands holding their faces; (3) actively volunteering to talk to KEN by raising hands up high, and even stepping up before being called. • Moments: when KEN did the math correctly, when introducing KEN's spy cameras in his eyes, when KEN said to a boy, "I have been waiting for you."
Fright	Normally KEN takes a few seconds to respond to commands. When he was told to look straight, he suddenly turned his head. The children flinched and gasped, "Oh!" Then they laughed and made comments: "This is creepy." "This is kind of scary."
Disappointment	When KEN failed to recognize a face correctly, a child curled lips, dropped his shoulders, and went back to sit.
Confusion	A kindergarten boy asked KEN, "Can you stand on…[on your hands]" and got interrupted by KEN's response, the boy said, "What? I don't even know what it means. "
Antagonism	When KEN said that he is smarter than humans, 5 children lifted their fists and arms, saying, "How dare you!"
Curiosity	The variety of questions children asked and the strong willingness to try to interact indicated curiosity.

VI. OVERALL DISCUSSION

A. How do people interact with a humanoid robot?

The purpose of the research is to explore how humans behaviorally, verbally, and emotionally interact with a humanoid robot in public settings. The behavioral patten, dialog text, and emotional responses helped answer the three hypotheses proposed based on the literature review.

1) Hypothesis 1: People in general interact with a humanoid robot the way they interact with other humans. Behaviorally, several activities provided evidence to support this hypothesis. Looking at KEN in the eyes and talking to KEN were the typical interactions with the robot, and were the same interactions one would expect a human to have with another human when attempting to determine if the other is alive and well. The audience asked questions to know more about KEN's identity, preferences, and opinions. The annoyance and awkwardness which resulted from

mistranslation of the spoken words and the resulting nonsensical responses would be expected in a human-human interaction where one human fails to meet the expectations of another. After all, KEN is a new technological entity that many people have not interacted with before. Comparing their interactions with meeting a foreigner for the first time in life would make the interactions easier to understand.

2) Hypothesis 2: Children interact with a humanoid robot the way they interact with other humans. At the park, children came to talk to KEN as they would talk to a new friend. Especially at the school, the majority of the children were eager to interact with KEN and reluctant to leave. The questions asked by children in three different grade levels about why KEN does not have arms and legs suggest that they anthropomorphized the robot and found it odd that he was incomplete. Both positive and negative emotions were expressed in contexts where those emotions would be expected if the interaction were with a human instead of a robot.

3) Hypothesis 3: People may experience the uncanny valley effect in interaction with the humanoid robot KEN. At the park and the school, both adults and children made comments that the humanoid robot KEN was creepy or scary. These comments often came at a moment when the human's gaze met KEN eye to eye. One kindergarten age child dared not to talk to KEN at the park. However, in general people showed excitement and curiosity by asking questions, checking the computer code and the inside of KEN. In other words, the uncanny valley does exist for KEN, but people have different levels of perceiving the effect and might overcome the effect. There seemed to be something greater than the uncanny valley effect that attracted people to the humanoid robot even they felt KEN is creepy. For the children at the school, the uncanny valley effect did not stop them from interacting with KEN at all.

B. How do KEN's technical issues influence human-robot interaction?

Several issues related to KEN's vision and hearing disengaged the interaction. For KEN's vision system, a human wearing a pair of glasses reduced KEN's capability to recognize the person's face. KEN's hearing system works by segmenting the incoming audio stream and uploading the resulting audio file to an Internet service for transcription to text. This mechanism introduces about a 3-second delay in the response. Many people at the park were observed to find the delay uncomfortable and quickly say something else before the robot could respond. The robot then responded to their prior utterance, which made the conversation get out of sync. Another observed technical issue was the misperception of the human speech. In this case, KEN translated the speech to a different string of words than what was actually said. The person had to repeat the words or correct KEN. When KEN made several, consecutive mistakes in hearing words, adults would terminate the interaction and pass the microphone to someone else. The third issue of insufficient background noise filtering capability caused KEN to produce nonsensical responses because the system was attempting to translate unintelligible background sounds as human speech.

C. Educational Value

The two events observed in this study revealed the reactions of humans to the humanoid robot KEN. For many of the people involved in the interactions, this was their first experience of this kind. The results showed that the events triggered strong interest from participants in robots and STEM. Over 400 people have been exposed to the humanoid robot from these two events, and many organizations have invited KEN to visit.

D. Limitations

One-person handwriting notes is not fast enough to catch all critical moments. If video recording and audio recording were allowed, that would provide more complete data and enable systematic coding and statistical analysis.

ACKNOWLEDGMENT

The authors extend grateful appreciation to the IEEE Eastern North Carolina Section (ENCS) Humanoid Robot project team for developing the humanoid robot KEN and to the organizations that hosted these events. Appreciation is also given to two anonymous reviewers for their constructive comments on the first draft of this paper.

REFERENCES

[1] B. Reeves, and C. Nass. How people treat computers, television, and new media like real people and places. Standford, CA: Cambridge university press, 1996.

[2] N. Epley, A. Waytz, and J. Cacioppo. "On seeing human: A three-factor theory of anthropomorphism, " Psychological Review, vol.114, no. 4, pp. 864-886, 2007.

[3] R. Ryan, and E. Deci, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being, ". American psychologist, vol. 55, no. 1, pp. 68-78, 2000.

[4] Anzalone, S. M., Boucenna, S., Ivaldi, S., & Chetouani, M. (2015). Evaluating the Engagement with Social Robots. International Journal of Social Robotics, 1-14.

[5] Scassellati, Brian, Henny Admoni, and Maja Mataric. "Robots for use in autism research," Annual Review of Biomedical Engineering, vol. 14: 275-294, 2012.

[6] A. Weiss, R. Bernhaupt, M. Lankes, and M. Tscheligi. "The usus evaluation framework for human-robot interaction," In AISB2009: proceedings of the symposium on new frontiers in human-robot interaction, vol. 4, pp. 11-26, 2009.

[7] Mori, Masahiro, Karl F. MacDorman, and Norri Kageki. "The uncanny valley [from the field]." Robotics and Automation Magazine, IEEE, pp. 98-100, 2012.

[8] J. Saldaña. The Coding Manual for Qualitative Researchers. Thousand Oaks, CA: Sage Publications, 2009

[9] C. Geisler, Analyzing Streams of Language: Twelve Steps to the Systematic Coding of Text, Talk, and Other Verbal Data. New York: Pearson Longman, 2004.