Article:

Hertzberg, J, Zhang, J, Zhang, L et al. (18 more authors) (2014) The RACE Project: Robustness by Autonomous Competence Enhancement. KI - Künstliche Intelligenz, 28 (4). pp. 297-304. ISSN 0933-1875

https://doi.org/10.1007/s13218-014-0327-y

Noname manuscript No.
(will be inserted by the editor)

The RACE Project

Robustness by Autonomous Competence Enhancement

Joachim Hertzberg · Jianwei Zhang · Liwei Zhang · Sebastian Rockel · Bernd Neumann · Jos Lehmann · Krishna S R Dubba · Anthony G. Cohn · Alessandro Saffiotti · Federico Pecora · Masoumeh Mansouri · Štefan Konečný · Martin Günther · Sebastian Stock · Luis Seabra Lopes · Miguel Oliveira · Gi Hyun Lim · Hamidreza Kasaei · Vahid Mokhtari · Lothar Hotz · Wilfried Bohlken

Received: date / Accepted: date

Abstract This paper reports on the aims, the approach, and the results of the European project RACE. The project aim was to enhance the behavior of an autonomous robot by having the robot learn from conceptualized experiences of previous performance, based on initial models of the domain and its own actions in it. This paper introduces the general system architecture; it then sketches some results in detail regarding hybrid reasoning and planning used in RACE, and instances of learning from the experiences of real robot task execution. Enhancement of robot competence is operationalized in terms of performance quality and description length of the robot instructions, and such enhancement is shown to result from the RACE system.

1 Project Aim and Demonstration Domain

RACE (Robustness by Autonomous Competence Enhancement) is a project funded by the European Commission under the 7th Framework Programme and running from 12/2011 to 11/2014. The partners are those institutes from which this paper is authored. This short project report summarizes the RACE methodology of working towards achieving these aims, and it sketches main project results, as visible about half a year before the end of the project.

The overall aim of RACE as set out in the description of work was

> to develop an artificial cognitive system, embodied by a service robot, able to build a high-level understanding of the world it inhabits by storing and exploiting appropriate memories of its experiences. Experiences will be recorded internally at multiple levels: high-level descriptions in terms of goals, tasks and behaviours, connected to constituting subtasks, and finally to sensory and actuator skills at the lowest level. In this way, experiences provide a detailed account of how the robot has achieved past goals or how it has failed, and what sensory events have accompanied the activities.

Contributions were foreseen in the description of work to advance the state of the art along three lines:

1. robots capable of storing experiences in their memory in terms of multi-level representations connecting actuator and sensory experiences with meaningful high-level structures,
2. methods for learning and generalising from experiences obtained from behaviour in realistically scaled real-world environments,
3. robots demonstrating superior robustness and effectiveness in new situations and unknown environments using experience-based planning and behaviour adaptation.

J. Zhang, L. Zhang, S. Rockel, B. Neumann, J. Lehmann
Hamburg University
E-mail: zhang@informatik.uni-hamburg.de

K. S. R. Dubba, A. G. Cohn,
University of Leeds

A. Saffiotti, F. Pecora, M. Mansouri, Š. Konečný
Örebro University

J. Hertzberg, M. Günther, S. Stock
Osnabrück University

L.Seabra Lopes, M.Oliveira, G.H.Lim, H.Kasaei, V.Mokhtari
University of Aveiro

L. Hotz, W. Bohlken
HITeC Hamburger Informatik Technologie-Center e. V.

Fig. 1 The PR2 robot Trixi grasping mugs from the counter.

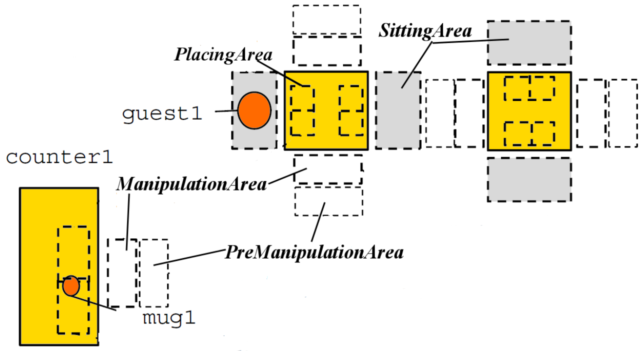

Fig. 2 Schema of an instance of the RACE demo scenarios in the restaurant domain. The counter of Fig. 2 is the `counter1` on the left. See text for more explanations.

So the thrust of the project was clearly of a conceptual nature. Yet, to demonstrate an integrated system and to have it learn from experiences, a physical robot and a demonstration domain are clearly needed. An "academic" demonstration domain was used to focus on the conceptual issues rather than application requirements, and to keep low the overhead for providing permanently the real-life demonstration scenario and for modeling it in a good simulation environment that would allow project partners to work independently at their sites between code integration events.

The demonstration domain is an AI and Robotics classic: a (mockup) restaurant with a robot waiter. The robot, Trixi, Fig. 2, is a PR2 with an additional RGB-D camera on top of its head. The task spectrum of Trixi is to serve guests in the mockup restaurant. Fig. 2 schematically shows one of a number of several scenarios defined for the restaurant domain; these scenarios are available both physically in a lab room and in simulation (Gazebo). Having such fixed scenarios allows tasks to be executed under somewhat controlled environment conditions to compare robot performances over different degrees of experience in the domain on Trixi's side.

In the sparsely populated scenario in Fig. 2, it would make sense to give Trixi the order "Serve mug1 to guest1!" or "Serve coffee to guest1!", both yielding the same service. It is also conceivable to teach Trixi how to serve: "Pick up the mug at the counter, bring it to the guest at table1 – this is how to serve a coffee".

It is assumed that basic robot behavior (such as navigation, object handling, object recognition) is available on Trixi – actually, RACE has started from standard capabilities available for a PR2 in ROS [17], cf. Sec. 2.1 for an explanation of the control architecture. Standard restaurant action schemata for a waiter, such as serving something to some guest, are available in a pre-defined form as Hierarchical Task Network (HTN) methods, cf. Sec. 3.1 for more on planning in RACE. Trixi was able to physically perform such restaurant standard actions in closed-loop plan-based control from early on in the project, based on the control architecture explained later (Sec. 2.1). This state-of-the-art approach was taken as the ground level of performance compared to which competence could be enhanced from experience by methods to be developed in the project.

Now what are reasons and opportunities for competence enhancement here? In a mundane domain like a restaurant, there is an infinite set of possibilities for variations of tasks to be executed in the light of actual conditions, even though the domain itself and the actions for a waiter (human or robot) to perform are highly schematized. These variations are the sources of possible disturbances for Trixi's execution – actually, they are the sources of the brittleness of autonomous robot performance in real-world settings that is so often deplored. They would in general result in non-nominal execution of the planned behavior, or in needed variations of the planned behavior at execution time. For example, unknown at planning time, paths may be blocked for the robot, the guest may have changed his seat on the table, standard placing areas on the table may be occupied by belongings of the guest, standard manipulation areas for the robot to stand while serving the table may be blocked, a newly arriving guest may interrupt plan execution, and so on. Conditions on all levels of description of robot performance (temporal, spatial, causal, perceptional, kinematic, dynamic) may actually deviate from the standard – no matter how the standard is formulated in detail. The RACE idea is that actually experiencing such deviations and learning ways how to deal with them (cf. Sec. 3.4) should lead to more robust performance in the domain. Moreover, being able to conceptualize such experiences and thereby to generalize them and make them amenable to the robot's own reasoning would result in a transfer from concrete experiences to classes of situations in which to change or adapt the standard behavior.

2 Approach

It is apparent from the overall aim of RACE that the project would face at least three methodological issues (which it shared with quite a few companion projects). First, a *bootstrapping problem*: to generate robot experiences to learn from, the project had to rely on a fully integrated and functional robot system in a suitable environment from the project start. Second, an *architecture problem*: to learn from conceptualized experiences of its own past behavior based not only on external features ("sensor streams"), but also on the internal control knowledge that led to generating the past behavior, all that data and knowledge has to be explicit and available for learning. Moreover, to be able to change its own behavior as a result of learning, the control knowledge yielding the behavior has to be explicit for the control. Third, an *evaluation problem*: to demonstrate competence *enhancement* after learning from experience, some performance metrics need to be used that would allow a sensible before-after comparison.

This section sketches the RACE solutions to these three issues.

The central point to solving the bootstrapping problem for RACE was early integration. The project has generated in its first year a fully integrated and functional robot in the restaurant domain with an initial instance of its target control architecture (cf. Sec. 2.1) in place. This was made possible by

- committing to a particular version of the above-described demo domain;
- using a PR2 robot and ROS as readily available hardware and software frameworks, respectively;
- using prior existing standard processing and reasoning modules as base systems wherever possible, e.g., for planning and sensor data interpretation;
- defining the internal knowledge-interchange language based on a standard, namely, Description Logics;
- and committing early to the basic robot control architecture, i.e., to a solution of the second problem addressed above.

Of these items, we will only detail the architecture issue, treated next; but we want to emphasize that the cross-topic and cross-workpackage results achieved in the project are to a large degree due to this early integration made possible in a joint effort by the partners.

The approaches to the control architecture and evaluation problems are described next in some detail.

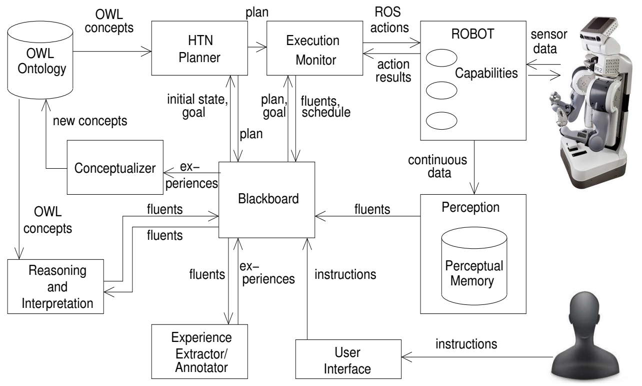

Fig. 3 The basic RACE architecture; modified from [20].

2.1 Control Architecture Approach

The cornerstone of the RACE architecture (Fig. 3) is the *Blackboard*. It mainly contains fluents, i.e., ground facts of the Description Logic (DL) ontology (executed actions, world state propositions, etc.), with begin and end timestamps. It is implemented as an RDF database. We decided to use a classical, "flat" blackboard in the project to allow for maximal flexibility of information flow between modules, including reasoning and learning modules, and for freely adding and exchanging versions of modules. This strategic advantage clearly comes at the cost of hard-wiring a bottleneck into the architecture; yet, the benefit has outweighed the cost in RACE.

The other modules for perception, reasoning, planning and execution communicate by reading selected types of information from the Blackboard, processing this information and writing back their outputs. So the Blackboard serves two roles: from the fluents on it, the current state as well as past state information can be derived; and it contains complete experience records, which can be conceptualized later.

When a new planning goal is entered by the user, an HTN Planner queries the Blackboard to build its initial planning state, then writes the generated plan back into the Blackboard. Initially, SHOP2 [13] was used, later replaced by the planner sketched in Sec. 3.1. The stored plan includes operators' preconditions and effects as well as the hierarchy of expanded HTN methods. The plan is picked up by the *Execution Monitor*, which dispatches the planned actions to the robot platform, mapping them to its closed-loop control modules. During execution, the monitor logs the executed actions, as well as success or failure information, in the Blackboard.

ROS [17], as used on Trixi, already provides many capabilities (e.g., for manipulation or navigation) as ROS actions; others were added. The robot provides continuous data about its own status (such as joint an-

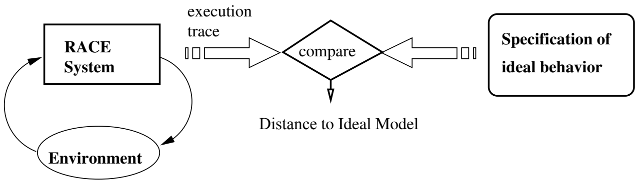

Fig. 4 Principle of evaluation: the system's behavior is compared to a model of the ideal behavior for the specific scenario.

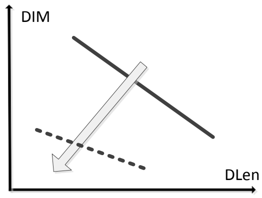

Fig. 5 RACE's aim: use as few as possible instructions (low DLen) to achieve correct behavior (low DIM). The enhancement of competence is indicated by the transition from the solid line to the dashed line

gles) as well as data from its sensors. The *Perception* module discretizes this information into symbolic, timestamped fluents.

The *OWL ontology* stores the robot's conceptual knowledge. It provides a common representation format, from which the knowledge used by all other reasoners is generated. Spatial, temporal, resource and ontological reasoners as well as a high-level scene interpretation module contribute higher-level semantic information to the experiences via the Blackboard.

Background processes responsible for experience extraction (grouped in an *Experience Extractor* module) and conceptualization (*Conceptualizer*) support a long-term learning loop, resulting in more robust and flexible future plans. The architecture is detailed in [20].

2.2 Evaluation Approach

To evaluate success for a given task in a given scenario, we measure the compliance of the actual robot behavior to the intended ideal behavior for that task in that scenario. Fig. 4 illustrates this principle: the trace of a given execution of Trixi is compared to a specification of what the ideal behavior should be, resulting in a "Distance to Ideal Model" (DIM) measure.

Discrepancies between the observed and the ideal behavior can originate from errors of four different types: **Conceptual**, **Perceptual**, **Navigation** and/or **Localization**, and **Manipulation** errors. The latter three types of errors are, to some degree, platform specific. Our metrics focus on quantifying conceptual errors.

Conceptual errors arise from discrepancies between the knowledge used by the robot and the one encoded in the specification of the ideal behavior. We call these discrepancies *inconsistencies*. Again, they can be of four types: (1) **Temporal**, (2) **Spatial**, (3) **Taxonomical**, and (4) **Compositional**. The DIM metric chosen in RACE is the weighted sum of the numbers of the inconsistencies (1–4), respectively, lower DIM values signaling better behavior.

In addition to estimating the effectiveness of learned knowledge by DIM, the *Description Length* (DLen, [19]) of the instructions given to the robot to achieve a goal matters. Normally, longer descriptions could yield better DIM as suggested by the solid line in Fig. 5. After learning from experiences, still successful or even more successful (even lower DIM) behavior following *shorter* instructions would be indicative of the effectiveness of the learned knowledge. This may indirectly provide a measure of how general the knowledge is, too, if applied to a wide range of scenarios and initial conditions.

So, the general RACE aim of designing learning and reasoning tools for a robot to autonomously and effectively increase its competence was operationalized as: make it possible for a robot to collect experiences allowing it to perform at lower DIM and shorter DLen.

3 Results

In addition to the overall system behavior, RACE has yielded a number of results in the individual modules shown in Fig. 3. They are sketched next. Details are in the references and on the website [18].

3.1 Hybrid Reasoning and Planning

To enable early integration as mentioned in Sec. 2, off-the-shelf planners were used in the beginning of the project. The goal was to analyze the limitations of the state of the art and develop an integrated planning system to overcome them. For task planning, HTN planning [5] proved to be useful for improving the robot's performance based on experience: the plan generation itself is fast, and the plans are robust and have a structure that can be used for learning.

While employing the off-the-shelf SHOP2 HTN planner was good for early integration, it was evident that state-of-the-art planning techniques were inadequate for the purposes of RACE: none of them could leverage the full knowledge that the project set out to learn from experience. The key issue is that this knowledge is hybrid addressing diverse semantics. For example, Trixi should serve `mug1` before the coffee gets cold, which requires reasoning in temporal knowledge. Similar arguments can be made about resource, spatial, causal, kinematic, and other forms of knowledge.

With the aim of building a planner that could leverage the different types of knowledge learned by the robot, we developed a general approach to hybrid reasoning: in a nutshell, it is based on a backtracking search algorithm that systematically explores the Cartesian product of sub-problems posed by the different fragments of knowledge possessed by the robot. Knowledge is represented as constraints (temporal, spatial, resource and causal relations), and the algorithm enforces the mutual feasibility of all constraints through backtracking and specialized hybrid reasoning procedures, called meta-constraints; the system for handling the constraint satisfaction problem (CSP) on these diverse knowledge levels is the Meta-CSP system [**?**]. Spatial knowledge in particular is represented in ARA^+, a novel spatial calculus that allows to uniformly account for metric and qualitative spatial knowledge in the sense-plan-act loop. Details of the hybrid planning approach and the KR formalisms used are presented in [11] (see also videos at [18]).

Although capable of combining task planning with other forms of reasoning, the approach alone does not leverage sophisticated planning heuristics, nor does it provide hierarchical decomposition capabilities in its domain specification language. Therefore, HTN hierarchization and and decomposition methods were put on top of the basic hybrid Meta-CSP planner, using the SHOP2 Total-order Forward Decomposition (TFD) algorithm for focusing search in the large combined search space.

The notion of using different types of knowledge at planning time was also leveraged for plan execution, through what is informally called a "semantic" execution monitor. This module continuously assesses plan feasibility in the light of additional information gathered during execution, and dispatches planned actions when their preconditions are fulfilled. As pointed out in [8], the planner's knowledge and that of the semantic execution monitor need not overlap completely: some of it may be execution-specific for improving robustness and enabling early failure detection.

3.2 Prediction

RACE uses the high-level scene interpretation system SCENIOR [2] for the robot to predict events and occurrences that may arise from a current situation. Trixi can envision possible developments of the environment as well as the impact that such developments may have on its own activities. The robot may use the results of such a prediction cycle to update the current situation before planning, thus producing more robust plans. For instance, based on its past experience, the robot may predict the likely presence of an obstacle at a relevant area (e.g., at one of the manipulation areas from which the guest can be served in Fig. 2). When planning its path towards the guest, the robot may therefore avoid such likely occupied areas. In this approach the robot has conceptual knowledge about occurrences in the world and about its own activities, represented in the ontology and in constraints expressed as Semantic Web Rule Language (SWRL) rules. Such conceptual knowledge is modeled or acquired by conceptualizing experiences collected in the robot's memory. Predictions are then generated by constructing models for explaining both incoming evidence and goals assigned to the robot as parts of the conceptual knowledge, this way generating possible future events, as detailed in [9]. Moreover, SCENIOR's prediction mechanism was enhanced with additional functionalities enabling the robot to extract not only the independent events from a given prediction but also to rank alternative predictions by their probability based on how frequently the predicted activities occurred in the robot's past experience. This will allow the system to focus on the causally relevant part of the most likely prediction during symbolic plan creation at planning time.

A second prediction approach in RACE [21] provides prediction during plan execution. It predicts non-nominal conditions, thereby improving system robustness. Non-nominal conditions in that sense are conditions that cannot be considered at planning time with purely symbolic planning approaches, such as robot manipulation failures, collisions, or object toppling events. This approach bases prediction upon commonsense physics, which is provided by the physics engine ODE used in Gazebo[1], the standard simulator in ROS. This allows detailed execution failures to be predicted, such as collision during manipulation or toppling of carried objects while the robot is moving fast in the environment. Based on the underlying simulation, this prediction is capable of delivering physics-based effects and results, thus generating possible future events, as well as simulated sensor data. To trigger prediction, so-called "imagination operators" are added to the planning domain. They are related to ordinary robot actions such as picking up an object and are predicted (executed in simulation) before being executed in reality. We refer to this prediction approach as *robot imagination*.

[1] http://gazebosim.org/

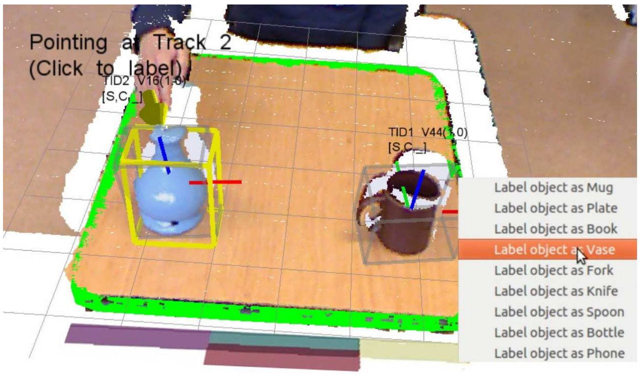

Fig. 6 The object perception and learning system interface.

3.3 Object Perception and Anchoring

The object perception system creates and maintains a representation of the locations of objects in the scene based on RGB-D data. This involves a set of capabilities organized as a pipeline:

– search the scene for objects not previously detected;
– visually track the detected objects to estimate their current poses;
– extract object features and recognize their categories based on learned knowledge;
– anchor perceived objects to symbolic instances in the blackboard.

The system must store object perception data as well as object category knowledge. However, the characteristics of perceptual information in general differ much from those of semantic information: while semantic information is symbolic and relational, perceptual data is typically numeric. To accommodate efficient storage and retrieval of both types of information, the RACE architecture features two memory systems working in parallel: a semantic memory system (the blackboard) and a perceptual memory system, cf. Sec. 2.1.

Three main design options address key computational issues involved in processing and storing perception data: a lightweight NoSQL database (leveldb) is used to implement the perceptual memory; a thread-based approach with zero copy transport of messages is used in implementing the modules; and a multiplexing scheme for processing different objects in the scene enables parallelization.

Object categories are learned with user mediation, as described in section 3.4, and stored in the perceptual memory. An anchoring module aggregates information from the object trackers into a probabilistic graphical model of all objects in the scene (including those not currently in view). Next, it uses probabilistic knowledge about typical geometric context between objects to jointly classify all objects. Finally, it updates the poses and object categories of objects on the blackboard to reflect the new maximum a-posteriori configuration of objects. In this way, object category symbols and object symbols used in semantic memory are grounded in the perceptual memory [4,16]. The RACE object perception system is fully integrated in the PR2. A video demonstrating this is available[2].

3.4 Learning

Learning is central to RACE, where the robot uses static and dynamic experiences to learn about static scenes, the environment, and its own activities for enhancing its competence to operate in its environment.

The human plays an important role in the robot's learning process. For instance, the human can teach categories of objects, methods for performing tasks and failure recovery strategies. An ontology of user instructions was defined and a simple user interface developed for this purpose. Experience extraction modules were developed to filter, segment and transform the raw data stream, producing experience records stored in memory. Segmentation and filtering of experiences are largely based on heuristics. In the case of supervised experience acquisition, experience extraction is triggered by teaching actions from the user [10]. These experiences are then conceptualized, leading to the formation and update of different concepts.

An *object conceptualizer* was developed to support the learning and recognition of object categories in an open-ended way [7,16]. This means that neither the categories nor the observations (experiences) that will support the learning are known in advance. Through pointing and labeling, a user triggers the extraction and recording of an object experience, and the respective conceptualization (see Fig. 6). An instance-based approach is adopted, in which an instance is stored when the robot fails to recognize its category. Recognition uses a nearest-neighbor approach with a distance measure normalized by an intra-category distance. Target objects too far from the known categories are judged to belong to an unknown category.

For more robust and flexible future robot task plans, an approach was developed to support the extraction and conceptualization of robot activity experiences [12]. After applying temporal segmentation heuristics, the experience data (a set of occurrences) is filtered using a graph simplification method based on ego networks [15]. Robot activity experiences are then conceptualized through deductive generalization, abstraction and feature extraction. The result is an *activity schema*

[2] http://youtu.be/XvnF2JMfhvc

that can be used as a method to solve future similar problems as well as a guide (or heuristic) to solve related (not strictly similar) problems.

To operate in an environment, it is important for the robot to understand the static *scene layouts*, as the scenes give the context for certain tasks, such as objects and activities to expect. It is also desirable for robots to be able to use human supervision and learn from different vague and incomplete input sources (perception, gestures, verbal and textual descriptions etc). In the RACE project, we approach this problem by converting data from different sources into relational format using spatial and temporal relations and then converting this data into graphs [4]. Information from different sources can be compared and contrasted using graph similarity measures to learn relational knowledge and models. The results obtained in the restaurant domain where we ground language in perception are encouraging, although some of the components (robust perception and natural language parser) need improvement for the whole system to be completely automatic.

The robot should also be able to recognize *environmental activities* so it can react appropriately. The RACE approach is based on qualitative and quantitative spatio-temporal features that encode the interactions between human subjects and objects in an abstract and efficient manner [3, **?**]. As a part of this research, we are constructing a semantically rich benchmark video dataset characterizing typical (simple and compound) environmental activities found in restaurants, which will be made public upon completion.

In RACE, robot activities are described by compositional hierarchies connecting activity concepts at higher abstraction levels with components at lower levels, down to action primitives of the robot platform. An obvious learning curriculum is therefore to let the robot construct new *compositional structures* based on existing activity concepts. In an approach described in detail in [14], Example-Based Compositional Learning (EBCL) is realized by constructing tentative concepts from examples and merging the concepts by computing a Good Common Subsumer (GCS), approximating a Least Common Subsumer (LCS, [1]). For example, a new concept "ServeACoffee" was constructed from services to guests at varying positions, performed with detailed instructions. EBCL suggests innovative solutions for at least three aspects: (i) Relevance analysis, i.e., determining objects and relations which are relevant for a new activity concept, (ii) a learning curriculum where positive examples lead to a learnt concept with monotonously increasing generality, never surpassing the intended concept, and (iii) a DL-based KR framework that can be mapped into graphical representations as used in the structure-mapping theory of Cognitive Science [6]. Besides learning from positive examples, EBCL also includes concept adaptation (generalizing a tentative concept to be applicable to a new situation) and concept refinement based on negative examples.

4 Summary of Achievements

The RACE project has developed, implemented and demonstrated in an integrated approach a robot control system able to improve its behavior by learning from conceptualizations of its own execution experiences. Central achievements include:

- a general approach for concurrently reasoning about diverse types of symbolic and metric knowledge, based on the notion of constraint reasoning at different levels of abstraction (Meta-CSP);
- Meta-CSP based algorithms for planning with domain specifications that include spatial, temporal, resource, causal and ontological knowledge;
- an approach to plan-based robot control that allows planning knowledge about deliberate robot behavior to be complemented by semantic execution monitoring and prediction;
- an object perception and learning system that learns object categories in an incremental and open-ended fashion with user mediation;
- an approach to learn conceptual activity descriptions from few examples and apply them to future tasks ("competence enhancement from experience");
- a method for grounding noun phrases connected by spatial relations in perceived static scenes.

To demonstrate an increase of robot competence, RACE has shown instances of DLen reduction by learning and of DLen and DIM reduction by handcrafted changes ("serve coffee" example). The final demonstrator, to be finalized after publication of this paper, will include instances of learned DLen and DIM reductions in the restaurant domain.

Acknowledgements RACE is funded by the EC FP7-ICT-2011-7, grant no. 287752, which is greatly appreciated.

Comments by the anonymous reviewers have helped improve this paper.

References

1. F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer wrt a background terminology. *J. Applied Logic*, 5(3):392–420, 2007.

2. W. Bohlken, P. Koopmann, L. Hotz, and B. Neumann. Towards ontology-based realtime behaviour interpretation. In H.W. Guesgen and S. Marsland, editors, *Human Behavior Recognition Technologies: Intelligent Applications for Monitoring and Security*, pages 33–64. IGI Global, 2013.
3. K.S.R. Dubba, A. G. Cohn, D.C. Hogg, M. Bhatt, and F. Dylla. Learning relational event models from video. *Submitted for publication*, 2014.
4. K.S.R. Dubba, M.R. de Oliveira, G.H. Lim, H. Kasaei, L. Seabra Lopes, A.M. Tomé, and A.G. Cohn. Grounding language in perception for scene conceptualization in autonomous robots. In *AAAI 2014 Spring Symposium on Qualitative Representations for Robots*, 2014.
5. K. Erol, J. Hendler, and D.S. Nau. HTN Planning: Complexity and expressivity. In *Proc. AAAI*, 1994.
6. D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170, 1983.
7. S.H. Kasaei, M. Oliveira, G.H. Lim, L. Seabra Lopes, and A.M. Tomé. An interactive open-ended learning approach for 3d object recognition. In *Proc. ICARSC*, 2014.
8. Š. Konecnỳ, S. Stock, F. Pecora, and A. Saffiotti. Planning domain+ execution semantics: A way towards robust execution? In *AAAI 2014 Spring Symposium on Qualitative Representations for Robots*, 2014.
9. J. Lehmann, B. Neumann, W. Bohlken, and L. Hotz. A robot waiter that predicts events by high-level scene interpretation. In *Proc. 6th Intl. Conf. Agents and Artificial Intelligence*, 2014.
10. G.H. Lim, M. Oliveira, V. Mokhtari, S.H. Kasaei, L. Seabra Lopes, and M.A. Tomé. Interactive teaching and experience extraction for learning about objects and robot activities. In *Proc. RO-MAN*, 2014 (accepted).
11. M. Mansouri and F. Pecora. More knowledge on the table: Planning with space, time and resources for robots. In *Proc. ICRA*, 2014.
12. V. Mokhtari, G.H. Lim, L. Seabra Lopes, and A.J. Pinho. Gathering and conceptualizing plan-based robot activity experiences. In *Proc. IAS*, 2014.
13. D. Nau, H. Muñoz-Avila, Y. Cao, A. Lotem, and S. Mitchell. Total-order planning with partially ordered subtasks. In *Proc. IJCAI*, 2001.
14. B. Neumann, L. Hotz, P. Rost, and J. Lehmann. A robot waiter learning from experiences. In *Proc. MLDM*, 2014.
15. M.E.J. Newman. Ego-centered networks and the ripple effect. *Social Networks*, 25(1):83–95, 2003.
16. M. Oliveira, G.H. Lim, L. Seabra Lopes, S.H. Kasaei, A. M. Tomé, and A. Chauhan. A perceptual memory system for grounding semantic representations in intelligent service robots. In *Proc. IROS*, 2014 (accepted).
17. M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A.Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
18. RACE – Robustness by autonomous competence enhancement. http://race.informatik.uni-hamburg.de/.
19. J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
20. S. Rockel and et al. An ontology-based multi-level robot architecture for learning from experiences. In *Designing Intelligent Robots: Reintegrating AI II, AAAI Spring Symposium*, 2013.
21. S. Rockel, D. Klimentjew, L. Zhang, and J. Zhang. An hyperreality imagination based reasoning and evaluation system (HIRES). In *Proc. ICRA*, 2014.
22. J. Tayyub, A. Tavanai, Y. Gatsoulis, A.G. Cohn, and D. Hogg. Qualitative and quantitative spatio-temporal relations in daily living activity recognition. Technical report, University of Leeds, 2014.