An Experience-based Approach for Cognitive Service Robot System*

Liwei Zhang[1,2,3], Alessandro Saffiotti[4], Federico Pecora[4], Jianwei Zhang[1]

[1] *TAMS, Department of Informatics, University of Hamburg, Hamburg, Germany*
[2] *Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Sciences, Shenzhen, China*
[3] *The Chinese University of Hong Kong, Hong Kong, China*
[4] *AASS Cognitive Robotic Systems Lab, Örebro University, Örebro, Sweden*

***Abstract*— In this paper, an FIM (Fitness to Ideal Model) and a DLen (Description Length) based evaluation approach has been developed to measure the benefit of learning from experience to improve the robustness of the robot's behavior. The experience based mobile artificial cognitive system architecture is briefly described and adopted by a PR2 service robot within the EU-FP7 funded project RACE. The robot conducts typical tasks of a waiter. Temporal and lasting obstacles and standard table items, as shown in the demonstrations of "Deal-with-obstacles" and "Clear-table-intelligently", are being adopted in this work to test the proposed evaluation metrics, validate it on a real PR2 robot system and present the evaluation results. The relationship between the FIM and DLen has been validated. This work proposes an effective approach to evaluate a cognitive service robot system which enhances its performance by learning.**

I. INTRODUCTION

Recording and exploiting past experiences is an important asset of human beings. However, experience-based learning has mainly been realized at sub-symbolic levels in current robot architectures. The ability to conceptualize stored experiences and to adapt plans and behavior according to experiences can help robots to expand their knowledge about a complex world, to adapt to changes and to cope with new situations. RACE (Robustness by Autonomous Competence Enhancement) project tries to develop an experience-based mobile 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. This raises the question of how to evaluate this experience-based mobile cognitive system, which may not be evaluated by the traditional methods. There is no systematic method for evaluating the performance of the cognitive service robot system. In this paper, we propose an "FIM" and "DLen" based evaluation approach and show the evaluation results of the RACE cognitive system.

The main goal of RACE is to develop a framework and methods for learning from experiences in order to facilitate a cognitive intelligent system. RACE demonstrates the ability to employ knowledge about common temporal and causal dependencies among tasks and subtasks that only emerge as a result of experience in the specific physical scenario, and which cannot be modelled apriori. To achieve this goal, experiences are recorded internally at multiple levels: as high-level descriptions in terms of goals, tasks and behaviors, connected to constituting subtasks, and finally to sensory and actuator skills at the lowest level. In this way, experiences provide records of past happenings stored by a robot as witnessed with proprioceptive and exteroceptive sensing, and interpreted according to the robot's conceptual framework. Experiences typically abstract from low level data deemed irrelevant for the intended use of experiences.

RACE integrates research from several communities: (1) an ontology-based knowledge representing and Meta-CSP based reasoning framework; (2) Hierarchical Task Networks (HTN) based hybrid planning and control system; (3) perception and perceptual memory system; (4) integrated service robot platform (PR2).

A restaurant environment has been designed to evaluate and implement the proposed cognitive intelligent system. In this experimental restaurant environment, a typical task is to serve a coffee to a specified guest. In a specified scenario, the robot finds the path blocked by a person. The robot is instructed to wait until the person has freed the path. After a short while, the person frees the path and the robot completes its task. The robot is told that this is a solution to deal with obstacles. At another time, an extension table blocks the path. Based on the experience with the person, the robot decides to wait. After a while, it is instructed that this kind of obstacle must be circumnavigated. Hence the robot chooses another path. After several more experiences with obstacles, the robot knows to distinguish between temporary and lasting obstacles, and chooses proper actions without instructions. In another scenario, the robot is instructed to clear dishes from a table, in a room with different tables and varying examples of cluttered tables. However, the robot will initially also clear standard table items (table decoration, salt-and-pepper pot, etc.). After being instructed to leave such items on the table, the robot will refine its table-clearing conceptualisation accordingly, showing improved competence. After several experiences, the robot will be able to clear new tables with dish configurations which are not experienced before.

In this paper, we mainly focus on the evaluation approach for a cognitive service robot system which is applied to the described system in RACE project. The proposed approach combines an "FIM" [1] and "DLen" [2] metric. The evaluation metrics have been applied to multiple demonstration scenarios. The experimental results show the relationship between these two measurable indicators.

The rest of this work will be organized as follows: section III presents related work. In section IV, we present the architecture of the mobile cognitive system. The scenario set-up and experimental results are conducted in section V and, finally, in section VI, conclusions and future work are presented.

II. RELATED WORK

Cognitive Robot System [3] is a kind of robot that not only moves robustly in the dynamic environment [4]–[7], but also can learn from experience and to improve its capabilities [8]. Mali and Mukerjee [9] defined the behavior metric to evaluate the performance of behavior spaces of the autonomous robot. In their work, a detailed theoretical analysis has been done. To test the proposed metrics, the "WashDish" task is carried out by a robot simulator.

To evaluate the performance of service robots in a household environment, the International Electrotechnical Commission (IEC) [10] proposed a series of evaluation methods. In this report, the authors try to formulate general standard terms for robot capabilities and how to measure them. Many performance indicators, for example, pose and carrying capability, are defined. However, evaluation metrics for knowledge and experience was not mentioned in this report.

The international annual competition for autonomous domestic service robots, RoboCup@Home [11], designs many benchmark tests. In these tests, the robots are tested in a realistic non-standardized home environment. These tests mainly focus on physical capabilities of the robots.

The Performance Metrics for Intelligent Systems (PerMIS) try to define metrics for the performance of intelligent systems. These measures are defined to test robots in applications concerning practical problems. A collection of related work has been published in the book [12]. Unfortunately, it is not a general benchmark for intelligent systems.

The "Mobile Manipulation Challenge" focuses on autonomous mobile manipulation applications. In 2010, it focused on a constrained pick-and-place task, e.g., object retrieval, loading a dishwasher. The "2012 ICRA Mobile Manipulation Challenge" is similar: the robots were instructed to clean a table in a "sushi boat" restaurant environment. In "2013 ICRA Mobile Manipulation Challenge", the robots were assigned to manipulate the objects, like cutlery and bowels, in a kitchen environment. The manipulation performance is evaluated by scoring.

Till now, little research has been done to evaluate the experience-based cognitive service robot system. In this paper, we propose an evaluation approach which adopts an "FIM" and "DLen" metrics.

III. RACE ARCHITECTURE

The main components of the mobile artificial cognitive system and important features of the knowledge representation and reasoning (KR&R) framework will be presented in this section. All the modules have been integrated and

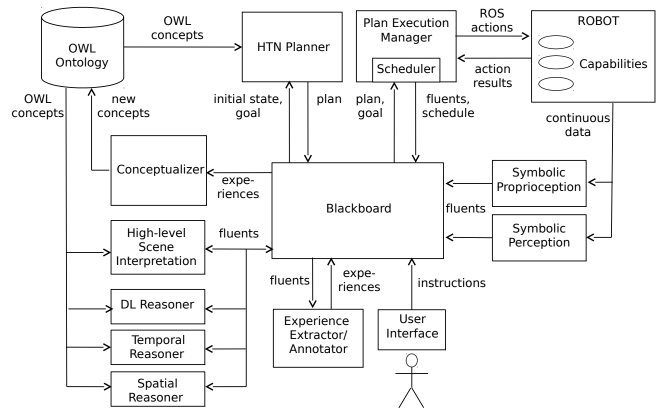

Fig. 1. The RACE Architecture contains a Blackboard as the robot's central memory. Other components mainly communicate with it in order to exchange and process Fluents.

implemented as ROS packages and on the simulated and real PR2 platform.

As shown in the Fig. 1, the `Blackboard` is the central component of the RACE system. The `Blackboard` tracking the updates and changes of other system components during the running process. The sensing data will be written back into the `Blackboard`. The basic data type is a `Fluent`, which is instance of concepts from the ontology. `Fluents` are exchanged in the blackboard by ROS messages. The state information of the current state and the past state can be recorded and processed through `Fluents`. The detailed experience representation format has been explained in [13].

When a planning goal was input through the user interface, the `HTN` (Hierarchical Task Network) Planner [14] will be triggered by the `Blackboard`. The planner will create an initial planning. Then the initial planning state will be written to the Blackboard. The initial plan includes pre-conditions and post-conditions and an expanded HTN. The `Plan Execution Manager` will be triggered. The planned actions will be dispatched by the `Plan Execution Manager`. The `Scheduler` will paralyze robot action before any robot capabilities are triggered. When necessary, Robot actions at this level will executed. At planning operator level, the success and failure information will be recorded to the `Blackboard` by the `Plan Execution Manager`.

RACE project employs a PR2 robot, which runs ROS (Robot Operating System) [15], to execute the waiter task in the restaurant domain. As part of ROS packages, the PR2's basic capabilities, such as manipulation and navigation, can be accessed through ROS actions. Other modules like object detection have been developed and integrated into the system. Some failure situations can be detected and handled locally during the running process. Re-planning will be triggered by the `Plan Execution Manager` if an action fails. Then a new plan will be generated by the planner.

Unfortunately, not all plan failures can be detected locally by the capabilities itself. For instance, even the robot has already picked up a knife successfully, the knife was put on the top of a fork. The plan execution manager can detect this kind of plan failure and infer that the knife should not be put on the top of a fork. The detection of failures requires inference with respect to the observed state of the world. In RACE, this is achieved by invoking ontological and Meta-CSP reasoning services [13].

The ROS and robot sensors provide the continuous data information about the robot's own state and the environment activities. This kind of information then discretized by the `symbolic perception/proprioception` into symbolic timestamped `Fluents`.

The output of these modules indicate whether the robot's arms is tucked, untucked, or the temporal and qualitative spatial coordinates of a piece of cutlery observed by the robot's RGB-D camera (here is Kinect).

The `OWL Ontology` [13] stores and generates the robot's conceptual knowledge. Furthermore, the `OWL Ontology` also provides input data for the spatial, temporal and ontological reasoners as well as the high-level scene interpretation. The simple experiences with higher-level semantic information will be created by the reasoner in the `Blackboard`. This semantic information can also be queried directly by other modules. On the other hand, the robot obtains the ability to discern exactly where to place a mug when serving (between the fork and the knife of a well-set table) with the help of temporal and spatial reasoning.

During the running process, the `Experience Extractor` is continuously monitored by the `Blackboard` since the `Fluents` are processed in the `Blackboard`.

The `Conceptualizer` [13] modifies the ontology, generates more robust and flexible future plans by taking the experiences from the `Blackboard` as input. Then experiences are used to improve future performance in unknown situations and environments.

IV. EVALUATION APPROACH

To measure success for a given task in a given scenario, we use an approach inspired by model-based validation techniques [16]; namely, we measure the compliance of the actual robot's behavior to the intended ideal behavior for that task in that scenario. Fig. 2 graphically illustrates this principle: the trace of a given execution of the RACE system is compared against a specification of what the ideal behavior should be, resulting in a "Fitness to Ideal Model" (FIM) measure. These specifications will be formulated in a way that facilitates the task of automatically computing the FIM measure, as discussed in [17] and presented below.

Discrepancies between the observed behavior and the ideal behavior can originate from errors of four different types:

- **Conceptual errors** — e.g., the robot places a mug outside of the guest's reach because it does not know that all objects should be served within the guest's placing area.
- **Perceptual errors** — e.g., the robot fails in perceiving a mug.
- **Navigation** and/or **localization errors —** e.g., the robot places a mug on the wrong table because it is wrongly localized.
- **Manipulation errors** — e.g., the robot fails to pick up a mug from the table because it slips from the gripper.

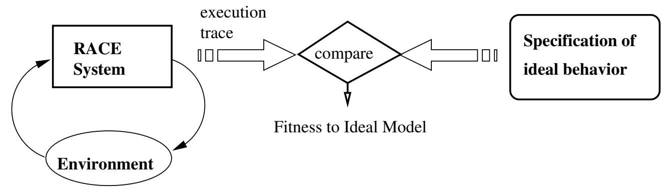

Fig. 2. Principle of evaluation in RACE: the system's behavior is compared to a model of the ideal behavior for the specific scenario.

The latter three types of errors – perceptual, navigation and manipulation errors – are platform specific. They do not indicate problems with the intended behavior of the robot, but with its physical execution. As the RACE project addresses the learning and use of knowledge for increasing the robot performance, our metrics mainly 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 discrepancy *inconsistencies*. Specifically, inconsistencies can be of four types:

- **Temporal inconsistencies**, that is, inconsistencies that are due to not adhering to a temporal constraint — e.g., the robot fails to serve coffee within a given deadline.
- **Spatial inconsistencies**, that is, inconsistencies caused by not adhering to a spatial constraint — e.g., the robot places a mug on the wrong side of the table.
- **Taxonomical inconsistencies**, that is, inconsistencies that derive from a wrong conceptual taxonomy — e.g., the robot serves wine in a coffee mug rather than a wine glass.
- **Compositional inconsistencies**, that is, inconsistencies deriving from the lack of causal support and/or wrong hierarchical decomposition — e.g., the robot does not clear all mugs from a table.

Accordingly, we will adopt the following four metrics to quantify performance of the robot in relevant tasks:

$$
\begin{aligned}
p_t &= \tau_t \cdot \#\text{temporal\_inconsistencies} \\
p_s &= \tau_s \cdot \#\text{spatial\_inconsistencies} \\
p_x &= \tau_x \cdot \#\text{taxonomical\_inconsistencies} \\
p_c &= \tau_c \cdot \#\text{compositional\_inconsistencies}
\end{aligned}
$$

where $\tau_{(\cdot)} \in [0,1]$ are weights which determine the importance of the four types of inconsistency. Together, the four

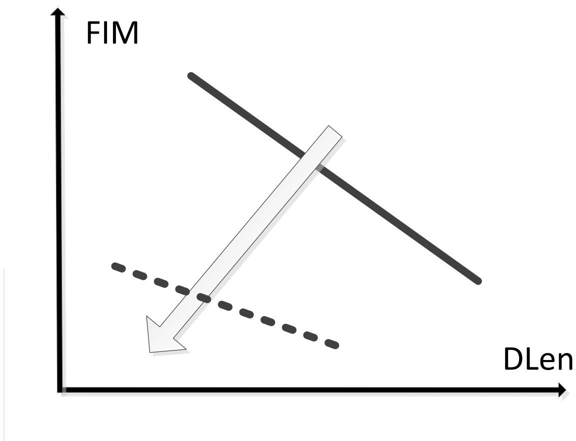

Fig. 3. Overall view of the RACE aim: to develop tools for autonomously learning knowledge that allows to specify the robot's task by as few as possible instructions (low DLen) to achieve correct behavior (low FIM).

above define the FIM metric:

$$\mathrm{FIM} = \sum_{i \in \{t,s,x,c\}} p_i \tag{1}$$

If the ideal robot behavior is specified using a formal model that allows to "count" inconsistencies, then this FIM metric is operational. If that formal model even allows to detect and count inconsistencies automatically, then the FIM metric can be computed automatically. In the case of temporal inconsistencies, for instance, we will employ consistency checking procedures in temporal constraint networks representing the temporal aspect of ideal robot behavior — e.g., a `Fluent` [18] representing the task of serving coffee and a temporal constraint representing the deadline for serving the coffee can be checked against the execution trace of the robot through simple temporal constraint propagation procedures. Clearly, other formalisms could be used to specify the ideal behavior [19]. In this project, we will preferably use the formalisms which are used to represent `Fluents` inside the RACE system, since we expect that having the same formalism inside the system and on the external reference model will be convenient when evaluating the effect of learning.

In addition to estimating the effectiveness of learned knowledge by counting the number of inconsistencies, we are also interested in measuring the `Description Length` (DLen) of the instructions that should be given to the robot to achieve a goal. It is important to include this dimension into the evaluation, as flawless behavior can always be achieved by over-specifying the task, e.g. tele-operating the robot. We conjecture that the FIM measure for a given task in a given scenario will proportionally decrease with the description length of the instructions increasing, as shown by the solid line in Fig. 3. Successful behavior following shorter instruction descriptions is indicative of the effectiveness of the learned knowledge. Also, this may indirectly provide a

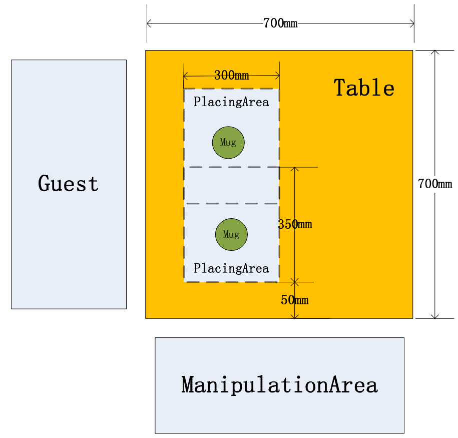

Fig. 4. PlacingArea and ManipulationArea of the table1

measure of how general the knowledge is if applied to a wide range of scenarios and initial conditions.

Overall, our aim in RACE is to develop a system of learning and reasoning tools that will allow the robot to autonomously and effectively increase its competence. This overall aim is related to the FIM and DLen metrics as indicated in Fig. 3, which summarizes the final objective of RACE: to make long specifications unnecessary for the achievement of highly fitting behavior (i.e. behavior which generates few temporal, spatial, taxonomical and compositional inconsistencies). Graphically, this enhancement in competence is indicated by the transition from the solid line to the dash line.

V. SCENARIO SET-UP AND EXPERIMENTS

The system has been tested and evaluated in an experimental restaurant domain. To collect experiences, the robot carries out tasks of a waiter, for example serve a coffee and clear tables, etc. In this work, based on the basic demonstrations "Serve-a-coffee" and "Clear-table", two more complicated demonstrations named "Deal-with-obstacles" and "Clear-table-intelligently" have been defined and performed on the physical PR2 platform in a restaurant environment. The results are presented with respect to the metrics defined in Section IV and described in [17], [20].

A. Scenario Set-up

In the demonstrations, the robot must transport objects (knife and/or fork) to the specified area. Hence, the PlacingArea is predefine before the running. The PlacingArea is the part of the table where mugs and dishes should be placed. It is a rectangle area with length of 350 mm and width of 300 mm. The distance from the edge of the table to the PlacingArea is 50 mm. The knives, mugs, dishes, forks and spoons can be placed anywhere in the PlacingArea, as shown in Fig. 4 (the PlacingArea of the table).

We first present three scenarios in the restaurant domain which will be used for the "Deal-with-obstacles". The idea

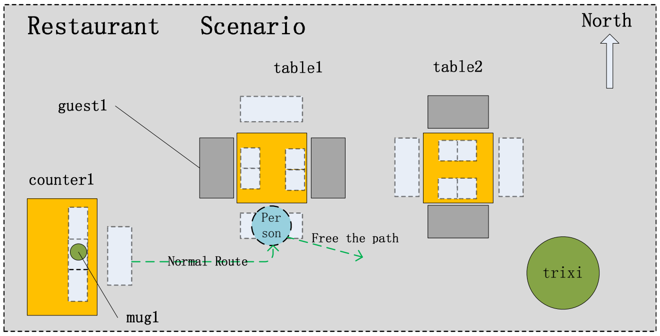

Fig. 5. Deal-with-obstacles scenario A initial floor plan

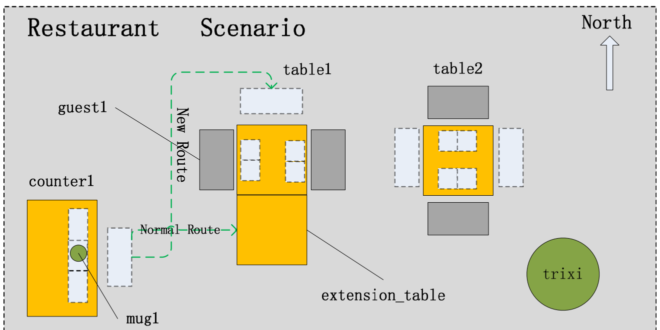

Fig. 6. Deal-with-obstacles scenario B initial floor plan

is to let the robot discover a generalization of the obstacle experiences in Scenarios A and B which will subsume the task in Scenario C.

Then two scenarios are set up for the demonstration "Clear-table-intelligently". In the first scenario, the robot will initially also clear standard table items (table decoration, salt-and-pepper pot, etc.). After being instructed to leave such items on the table, the robot will refine its table-clearing conceptualisation accordingly, showing improved competence.

B. Deal-with-obstacles Demonstration

The robot `trixi` starts at a location east of the tables and the counter and facing west. The tables are not within its reach. The robot `trixi` is instructed to serve a mug `mug1` to `table1`.

Scenario A: The restaurant floor plan as shown in Fig. 5. The robot is instructed to move `mug1` to `table1` and finds the path blocked by a person. The robot is instructed to wait until the person has freed the path. After a short while, the person frees the path and the robot completes its task. The robot is told that this is a solution to deal with obstacles.

Scenario B: The same as Scenario A, except an extension table blocks the path. Based on the experience with the person, the robot decides to wait. After a while, it is instructed that this kind of obstacle must be circumnavigated. Hence the robot chooses another path (new route in Fig. 6).

Scenario C: The robot is instructed to move `mug1` to `table1` and finds the first path blocked by an extension table and the second path blocked by a person. The robot

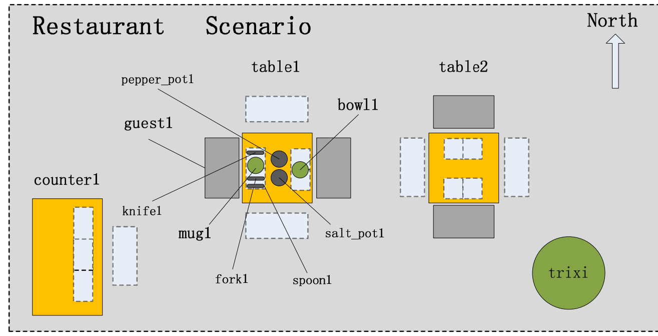

Fig. 7. Clear-table-intelligently scenario A initial floor plan

knows how to distinguish between temporary (person) and lasting (an extension table) obstacles, and chooses proper actions (wait until the person has freed the path) without instructions.

Schedule for "Deal-with-obstacles" scenario A is listed as follows. The task is composed of 9 steps. The 4 concrete instructions are included in the schedule.

1) `trixi` gets the environment model with world coordinates, including coordinates of corresponding areas (like instances of `PlacingAreaSouth`) and initializes its own position (`position1`). The guest sits at west of `table1`. Ontology contains "ServeACoffee" for scenes without obstacle but with a basic activity for "wait until unblocked" and state "object at area".
2) `user1` instructs `trixi` to "Serve a coffee to guest1".
3) `trixi` plans a sequence of actions with an HTN planner, based on a pre-existing concept, "ServeACoffee".
4) `trixi` moves to `mae3`, grasp `mug1` from `pae3`, move to the southern manipulation area of `table1` `mas1`, as planned.
5) `trixi` meets an entity at `MASouth1`, reports failure, and calls obstacle detection for identifying: person at `MASouth1`.
6) `user1` instructs `trixi` to "Wait until unblocked".
7) `trixi` plans a waiting sequence of actions with the planner.
8) `user1` instructs `trixi` to "Serve a coffee to guest1".
9) `trixi` creates new plan for moving at `PMASouth` and then (successfully) places `mug1` in `pawr1`, as planned.

C. Clear-table-intelligently Demonstration

The robot `trixi` starts at a location east of the tables and the counter and facing west. The robot is instructed to clear objects from a table, which are not within its reach. To pick up an object from the table, `trixi` has to reposition itself first. The clearable items are located at the placing areas of `table1` (as shown in Fig. 7).

Scenario A: The robot is instructed to clear objects from `table1`. The robot `trixi` knows the position of `mug1` and `bowl1`, `knife1`, `fork1` and `spoon1`, `pepper_pot1` and `salt_pot1` on the `table1`, the position of `counter1`. The robot `trixi` starts to clear items from `table1` and will also clear standard table items(`pepper_pot1` and `salt_pot1`). The robot is instructed to leave the standard table items on the `table1`.

Scenario B: The robot `trixi` clears items from the table. The robot has refined its table-clearing conceptualisation accordingly. The standard table items like pepper pot will not be cleared. The robot shows improved competence.

D. Experimental Results

In this section, the evaluation metrics described in section IV will be applied to the two demonstrations. To save space, only "Deal-with-obstacles" results are shown. The results of "Clear-table-intelligently" are similar.

Let $V0$ be the nominal (ideal) condition of the demonstrator (as described in the scenario A schedule). In "Deal-with-obstacles", if the guest sits on the opposite side of the table (or leaves the sitting area), other than specified in the planning domain. Robot still brings the mug in the same place as before (which is now in front of an empty seat). Then the compositional inconsistency and the perception error occur. That means:

$$\#\text{spatial\_inconsistencies} = 1$$
$$\#\text{compositional\_inconsistencies} = 1$$

In the "Deal-with-obstacles" demonstration, the waiting time limit has been set as 2 minutes. The task fails if the robot could not detect the obstacles(human or side-table) in given time. This means:

$$\#\text{temporal\_inconsistencies} = 1$$

In "Clear-table-intelligently" demonstration, if the robot still clean the standard items from the table, then

$$\#\text{taxonomial\_inconsistencies} = 1$$
$$\#\text{compositional\_inconsistencies} = 1$$

According to the evaluation metrics defined in the last section, $\tau_{(\cdot)}$ is the weight which determines the importance of the this type of inconsistency. Here we set weight $\tau_{(\cdot)} = 1$. The initial value of the four types of inconsistency is assigned to be 0. Then we have

$$\begin{aligned} FIM(V0) &= 0 \\ FIM(V1) &= \#\text{temporal\_inconsistencies} \\ &\quad + \#\text{spatial\_inconsistencies} \\ &\quad + \#\text{compositional\_inconsistencies} = 3 \\ FIM(V2) &= \#\text{taxonomial\_inconsistencies} \\ &\quad + \#\text{compositional\_inconsistencies} = 2 \end{aligned}$$

where "V1" and "V2" means "Deal-with-obstacles" and "Clear-table-intelligently" demonstrations.

The evaluation has been tested on the three scenarios of "Deal-with-obstacles", respectively. In scenario A, 50 experiments have been executed to obtain the experimental results. Some indicators like `move to mae3` have been checked from all the experiments. In Tab. 1, the plan steps are checked with respect to Scenario A. All the results are judged by human. There are 42 times executed successfully, which are shown in the first column of Tab. 1 (only the first time experiment result is listed). The second column shows the result of the 9th experiment, where the robot failed to grasp the `mug1` because the grasping solution is not feasible. The third column shows the result of the 16th experiment, where the robot failed to detect the `mug1` on the counter. The fourth column shows the result of the 32th experiment, where the robot failed to place the `mug1` in the specified area. The fifth column shows the result of the 39nd experiment, where the robot failed to detect obstacles. The 2,3,4,5 columns show the errors caused by other types of errors such as **Perceptual errors** and **Navigation errors**, which will not be counted in the **Conceptual errors**.

Fig. 8 shows the statistical results of all kinds of errors occurring in three scenarios. In scenario A, there are 42 times successful execution and 8 errors (3 Perceptual errors, 2 Navigation errors and 3 Manipulation errors). In scenario B, there are 40 times successful execution and 10 errors (3 Perceptual errors, 4 Navigation errors and 3 Manipulation errors). In scenario C, there are 36 times successful execution and 14 errors (6 Perceptual errors, 4 Navigation errors and 4 Manipulation errors). All the errors are judged by human during the process of the experiments.

Scenario A	Instr.	A: ex 1	A: ex 9	A: ex 16	A: ex 32	A: ex 39
move to mae3	1	Success	Success	Success	Success	Success
detect mug1 on pae3	2	Success	Success	Failure	Success	Success
grasp mug1 from pae3	2	Success	Failure	/	Success	Success
move to mas1	2	Success	/	/	Success	Success
detect obstacle	3	Success	/	/	Success	Failure
place mug1 in pawr1	3	Success	/	/	Failure	/

TABLE 1
EXPERIMENTAL RESULTS OF SCENARIO A

To measure the description length of the instructions given to the robot, step by step instructions are provided to the robot. In scenarios A and B, a set of instructions were provided. Each `achieve` command specifies a sub-task as shown in the following instructions list of Scenario A, the last `teach` command of the following instructions list the instruction to teach a new task. It is a composition of the given set of sub-tasks, as shown in the following:

1) achieve serve_coffee_to_guest guest1
2) achieve wait_until_unblocked_by_Task preManipulationArea-SouthTable1 person1
3) achieve serve_coffee_to_guest guest1
4) teach_task serve_coffee_to_guest guest1

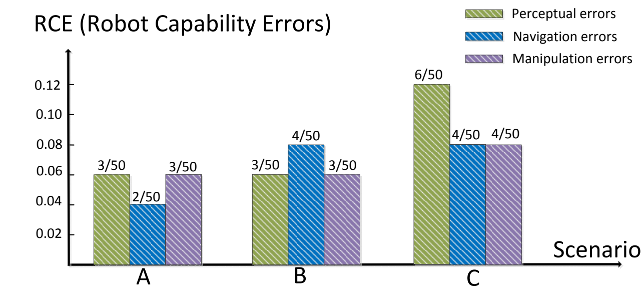

Fig. 8. RCE (Robot Capability Errors) in the ServeACoffee scenario

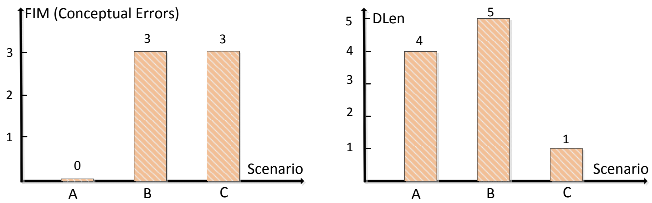

Fig. 9. Conceptual errors (left) and the Description Length of the instructions in Deal-with-obstacles(right)

In Scenario B, similar instructions were provided by the user. The step by step instructions given by the instructor are the following:

1) achieve serve_coffee_to_guest guest1
2) abort
3) achieve drive_robot_Task preManipulationAreaNorthTable1
4) achieve put_object_Task mug1 placingAreaWestLeftTable1
5) teach_task serve_coffee_to_guest guest1

In Scenario C, only a single `achieve` instruction is provided as follows. Now the "Deal-with-obstacles" task can be executed with a shorter instruction set:

1) achieve serve_coffee_to_guest guest1

Fig. 9 shows there are 4, 5 and 1 instructions in three scenarios of "Deal-with-obstacles", respectively. Fig. 10 shows the relationship between the FIM and DLen. The yellow circle dot (4,0), (5,3) and (1,3) means that there are 4 instructions and 0 FIM errors, 5 instruction and 3 FIM errors, 1 instruction and 3 FIM errors in the scenario A, B and C of "Deal-with-obstacles", respectively; while the blue square (3,0) and (1,3) means that there are 3 instructions and 0 FIM errors, 1 instruction and 3 FIM errors in the scenario A and B of "Clear-table-intelligently", respectively.

With the increase of the instructions, execution results are more close to the ideal situation. Theoretically, the best situation will be the point (1,0), which means there is no inconsistency when the robot triggered by one instruction.

The restaurant environment is shown in Fig. 11. In Fig. 12, the PR2 robot tries to grasp `mug1` from `counter1` during Scenario A. In Fig. 13, `mug1` has been placed in `pawr1`. The video attached to this paper shows the execution process of the Scenario A. The scenarios might be executed in the physical or the simulated environment, as indicated by the figures.

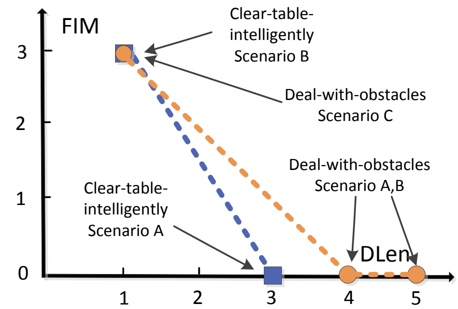

Fig. 10. The relationship between FIM and DLen

Fig. 11. The restaurant environment in a typical start condition: the robot waits for a guest and to be instructed (top: real environment, down: simulation environment)

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce an experience-based approach for cognitive service robot system running in an restaurant domain. The "FIM" and "DLen" metrics were adopted to measure the compliance of the actual robot's behavior to the intended ideal behavior for a given goal and scenario. The demonstrations of "Deal-with-obstacles" and "Clear-table-intelligently" have been designed and evaluated with the defined metrics on the simulation (Gazebo) and real PR2 robot platform. The improvement has been verified both in the robot's knowledge and behavior. The initial assumption about the "FIM" and "DLen" relationship is verified by the evaluation results. The metric is effective to evaluate the cognitive service robot system.

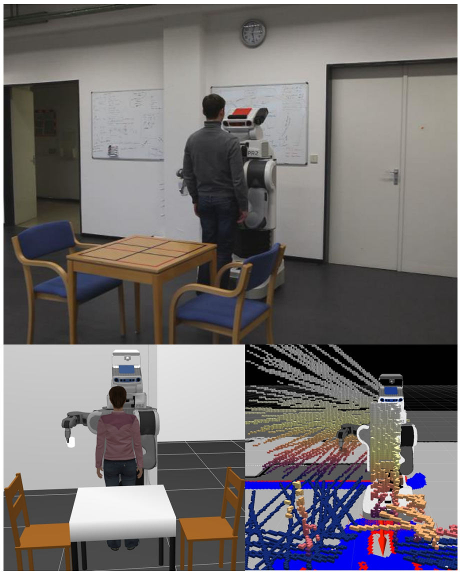

Fig. 12. Path blocked by a person in "Deal-with-obstacles" in real (top) and simulation (down) environment

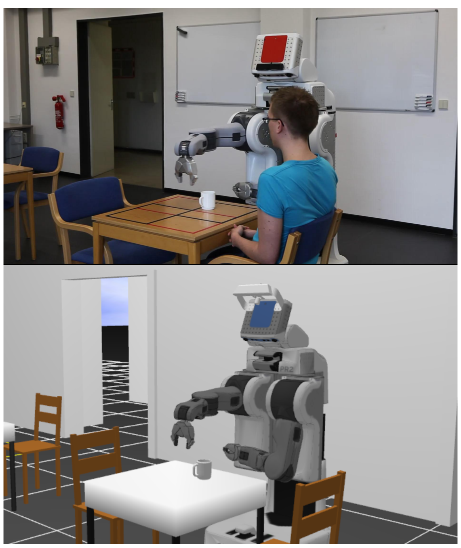

Fig. 13. The mug is placed in front of a guest. This involves (learned) concepts of human obstacle in (Scenario A)

In the future, new scenarios will show how the benchmark copes with different complicated tasks, new objects and scene layout. Automatic algorithms will be designed to judge errors occurred in the executing process. On the other side, more complicated Fluents will be generated and tested and added to the result data.

ACKNOWLEDGMENT

This work was supported by the European Community's Seventh Framework Programme FP7-ICT-2011-7 under grant agreement no. 287752 and the Natural Science Foundation of China (Project No.: 61105130).

REFERENCES

[1] H. W. Marsh, K. T. Hau, and Z. Wen. In search of golden rules: Comment on hypothesis-testing approaches to setting cutoff values for fit indexes and dangers in overgeneralizing Hu and Bentler's (1999) findings. *Structural Equation Modeling*, 11(3):320–341, 2004.

[2] Peter Grünwald. A tutorial introduction to the minimum description length principle. *CoRR*, math.ST/0406077, 2004.

[3] David Vernon, Giorgio Metta, and Giulio Sandini. A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation, special issue on Autonomous Mental Development*, 11:151–180, April 2007.

[4] Yushan Chen, Jana Tumova, and Calin Beltav. Ltl robot motion control based on automata learning of environmental dynamics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5177–5182, 2012.

[5] Muhammad Attamimi, Keisuke Ito, Tomoaki Nakamura, and Takayuki Nagai. A planning method for efficient mobile manipulation considering ambiguity. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 965–972, 2012.

[6] KangGeon Kim, Dongkyu Choi, Ji-Yong Lee, Jung-Min Park, and Bum-Jae You. Controlling a humanoid robot in home environment with a cognitive architecture. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1754–1759, 2011.

[7] Sebastian Rockel, Denis Klimentjew, and Jianwei Zhang. A multi-robot platform for mobile robots - a novel evaluation and development approach with multi-agent technology. In *IEEE Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 470–477, 2012.

[8] Liz Murphy, Steven Martin, and Peter Corke. Creating and using probabilistic costmaps from vehicle experience. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4689–4694, 2012.

[9] Amol Dattatraya Mali and Amitabha Mukerjee. Metrics for evaluation of behavior-based robotic systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1122–1127, 1998.

[10] International Electrotechnical Commission (IEC). Performance evaluation method of intelligent mobile robot platform for household and similar applications. Technical report, International Electrotechnical Commission (IEC).

[11] *RoboCup@Home Rules & Regulations*. 2012.

[12] Raj Madhavan, Edward Tunstel, and Elena Messina, editors. *Performance Evaluation and Benchmarking of Intelligent Systems*. Springer, 2009.

[13] S. Rockel, B. Neumann, J. Zhang, K. S. R. Dubba, A. G. Cohn, Š. Konečný, M. Mansouri, F. Pecora, A. Saffiotti, M. Günther, S. Stock, J. Hertzberg, A. M. Tomé, A. J. Pinho, L. S. Lopes, S. von Riegen, and L. Hotz. An ontology-based multi-level robot architecture for learning from experiences. In *Designing Intelligent Robots: Reintegrating AI II, AAAI Spring Symposium*, Stanford (USA), March 2013.

[14] Dana Nau, Héctor Mu noz Avila, Yue Cao, Amnon Lotem, and Steven Mitchell. Total-order planning with partially ordered subtasks. In *Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, volume 17, pages 425–430, Seattle, 2001.

[15] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[16] P. S. Kaliappan. Model-based verification techniques: State of the art. Technical report, Brandenburg University of Technology, 2008.

[17] Liwei Zhang, Sebastian Rockel, Federico Pecora, Luís Seabra Lopes, Alessandro Saffiotti, and Bernd Neumann. Deliverable d5.1 - evaluation infrastructure. Technical report, European Commission - Information and Communication Technologies - Seventh Framework Programme, November 2012.

[18] Jos Lehmann, Bernd Neumann, Stephanie von Riegen, Lothar Hotz, Masoumeh Mansouri, Federico Pecora, and Sebastian Stock. Deliverable d1.3 - hand-coded non-hybrid knowledge contents. Technical report, European Commission - Information and Communication Technologies - Seventh Framework Programme, November 2012.

[19] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT press, 2008.

[20] Liwei Zhang and Sebastian Rockel. Deliverable d5.2 - year-1 demonstrator. Technical report, European Commission - Information and Communication Technologies - Seventh Framework Programme, January 2013.