The Assistive Kitchen —
A Demonstration Scenario for Cognitive Technical Systems

Michael Beetz, Freek Stulp, Bernd Radig,
Jan Bandouch, Nico Blodow, Mihai Dolha, Andreas Fedrizzi, Dominik Jain, Uli Klank, Ingo Kresse,
Alexis Maldonado, Zoltan Marton, Lorenz Mösenlechner, Federico Ruiz, Radu Bogdan Rusu, Moritz Tenorth
Intelligent Autonomous Systems, Technische Universität München, Department of Informatics
Boltzmannstr. 3, D-85748 Garching bei München, Germany

***Abstract*— This paper introduces the *Assistive Kitchen* as a comprehensive demonstration and challenge scenario for technical cognitive systems. We describe its hardware and software infrastructure. Within the *Assistive Kitchen* application, we select particular domain activities as research subjects and identify the cognitive capabilities needed for perceiving, interpreting, analyzing, and executing these activities as research foci. We conclude by outlining open research issues that need to be solved to realize the scenarios successfully.**

I. INTRODUCTION

Cognitive technical systems are systems that are equipped with artificial sensors and actuators, integrated into physical systems, and act in a physical world. They differ from other technical systems in that they have *cognitive capabilities* including perception, reasoning, learning, and planning that turn them into systems that *"know what they are doing"* [7]. The cognitive capabilities will result in systems of higher reliability, flexibility, adaptivity, and better performance and systems that are easier to interact and cooperate with.

The cluster of excellence COTESYS (Cognition for Technical Systems) [8] considers the assistance of elder people to be a key application where technical cognitive systems could profoundly impact the well-being of our society. Therefore, COTESYS investigates the realization of an *assistive kitchen* (Fig. 1), a ubiquitous computing, sensing, and actuation environment with a robotic assistant as one of its primary demonstration scenarios. The Assistive Kitchen aims at

- supporting and assisting people in their household chores through physical action;
- enhancing the cognitive capabilities of people doing household work by reminding them; and
- monitoring health and safety of the people.

To achieve these objectives, the Assistive Kitchen is to

- perceive, interpret, learn, and analyze models of household chore and activities of daily life (ADLs); and
- represent the acquired models such that the Assistive Kitchen can use them for activity and safety monitoring, health assessment, and for adapting itself to the needs and preferences of the people.

The research reported in this paper is partly funded by the German cluster of excellence COTESYS (Cognition for Technical Systems). More information including videos and publications about the *Assistive Kitchen* can be found at `http://ias.cs.tum.edu/assistivekitchen`. Due to space limitations this paper does not give a comprehensive discussion of related work.

The Assistive Kitchen includes an autonomous robotic agent that is to learn and perform complex household chores. The robot must perform housework together with people or at least assist them in their activities. This requires safe operation in the presence of humans and behaving according to the preferences of the people they serve.

Clearly, assistive kitchens of this sort are important for several reasons. First, they are of societal importance because they can enable persons with minor disabilities including sensory, cognitive, and motor ones to live independently and to perform their household work. This will increase the quality of life as well as reduce the cost of home care.

Assistive kitchens and living environments also raise challenging research problems. One of these problems is that performing household chores is a form of everyday activity that requires extensive commonsense knowledge and reasoning [3]. Another challenge is the low frequency of daily activities, which requires embedded systems and robotic agents to learn from very scarce experience. Besides, household chores include a large variety of manipulation actions and composed activities that pose hard research questions for current robot manipulation research. The management of daily activities also requires activity management that is very different from that commonly assumed by AI planning systems.

II. ASSISTIVE KITCHEN INFRASTRUCTURE

We start with the hardware and software infrastructure of the kitchen — the implementational basis that defines the possibilities and restrictions of the demonstration scenarios.

A. The Hardware Infrastructure

The hardware infrastructure consists of a mobile robot and networked sensing and actuation devices that are physically embedded into the environment.

1) The Autonomous Mobile Robot: Currently, an autonomous mobile robot with two arms with grippers acts as a robotic assistant in the Assistive Kitchen (see Fig. 1). The robot is a RWI B21 robot equipped with a stereo CCD system and laser rangefinders as its primary sensors. One laser range sensor is integrated into the robot base to allow for estimating the robot's position within the environment. Small laser range

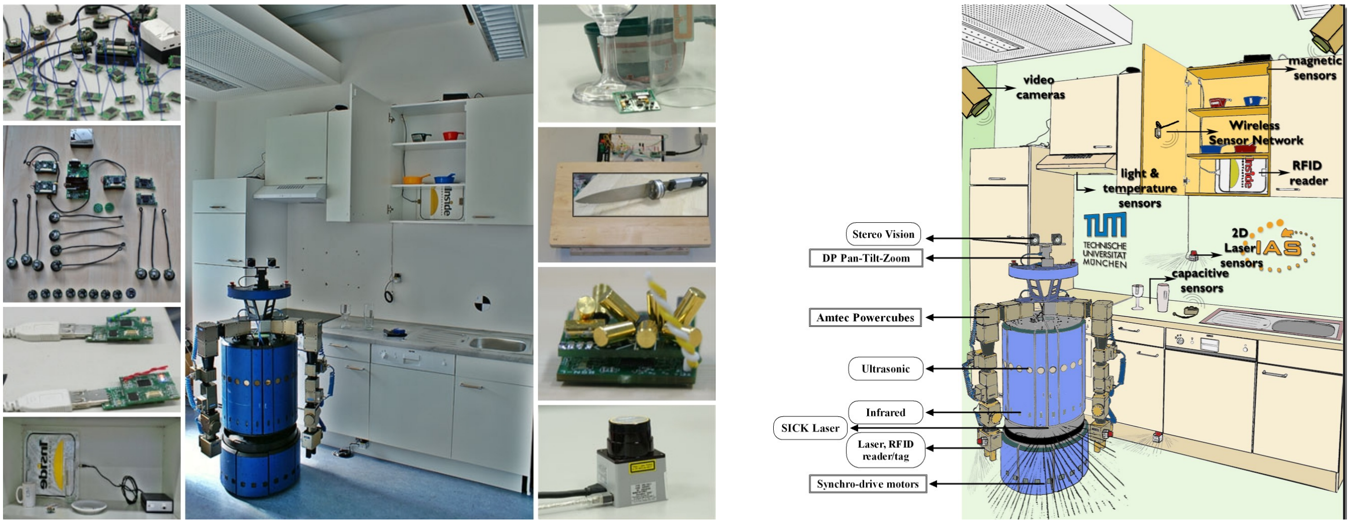

Fig. 1. The Assistive Kitchen containing a robot and a variety of sensors.

sensors are mounted onto the robot's grippers to provide sensory feedback for reaching and grasping actions. The grippers are also equipped with RFID tag readers that support object detection and identification. Cameras are used for longer range object recognition and to allow for vision-based interaction with people. This robot will be complemented by the Justin robot for sophisticated manipulation tasks.[1]

2) Room Infrastructure: The sensor-equipped kitchen environment is depicted in Fig. 1. In this kitchen, a set of static off-the-shelf cameras is positioned to cover critical working areas with high resolution. With these cameras, human activity and robots can be tracked from different locations to allow for more accurate positioning and pose estimation (see Section IV-B). In addition, laser range sensors are mounted at the walls for covering large parts of the kitchen. They provide accurate and valuable position data for the people present in the environment and their movements within the kitchen.

The pieces of furniture in the Assistive Kitchen are also equipped with various kinds of sensors. RFID tags and magnetic and capacitive sensors provide information about the objects in a cupboard or on the table, and whether a cupboard is open or closed.

Furthermore, small ubiquitous devices offer the possibility to instrument people acting in the environment with additional sensors. We have equipped a glove with an RFID tag reader that enables us to identify the objects that are manipulated by the person wearing it. In addition, the person is equipped with tiny inertial measurement units that provide us with detailed information about the person's limb motions. Another body worn sensory device to be used in the Assistive Kitchen demonstration scenario is the gaze-aligned head mounted camera which allows the estimation of the attentional state of people while performing their kitchen work.[2] These last two sensors will be presented in Section IV-B.

Web-enabled kitchen appliances such as the refrigerator, the oven, the microwave, and the faucet, allow for remote and wireless monitoring and control.

B. *Software Infrastructure*

A critical factor for the successful implementation of the assistive kitchen is the software infrastructure. It has to provide a simple, reliable, uniform, and flexible interface for communicating with and controlling different physically distributed sensors and actuators.

We use and extend the open-source Player/Stage/Gazebo (P/S/G) software library to satisfy these requirements for the sensor-equipped environment as well as the robotic agent. Player provides a simple and flexible interface for robot control by making available powerful classes of interface abstractions for interacting with robot hardware, in particular sensors and effectors. These abstractions enable the programmer to use devices with similar functionality with identical software interfaces, thus increasing the code transferability. An enhanced *client/server model* featuring auto-discovery mechanisms as well as permitting servers and clients to communicate between them in a heterogeneous network, enables programmers to code their clients in different programming languages. Programmers can also implement sophisticated algorithms and provide them as Player drivers. By incorporating well-understood algorithms into our infrastructure, we eliminate the need for users to individually re-implement them [10].

Using the P/S/G infrastructure, a robot can enter a sensor-equipped environment, auto-discover the sensors and the services they provide and use these sensors in the same way as it uses its own sensors.

[1]The Justin robot is currently under development at the DLR robotics institute: `http://www.robotic.dlr.de`

[2]The gaze-aligned head mounted camera is currently developed by Neurologische Klinik und Poliklinik, Ludwig-Maximilians-Universität München, which participates in the CoTeSys excellence cluster (`http://www.forbias.de/`).

Figure 2 depicts the computational structure of the controller of the household robot. The control system is composed of two layers: the *functional* and the *task-oriented layer*. At the functional layer the system is decomposed into distributed modules where each module provides a particular function. A function could be the control over a particular actuator or effector, access to the data provided by a particular sensor subsystem, or a specific function of the robot's operation, such as guarded motion. The functional layer is decomposed into the perception and the action subsystem. The modules of the perception subsystem perceives states of the environment and the modules of the action subsystem control the actuators of the robot. All these modules are provided by our P/S/G infrastructure.

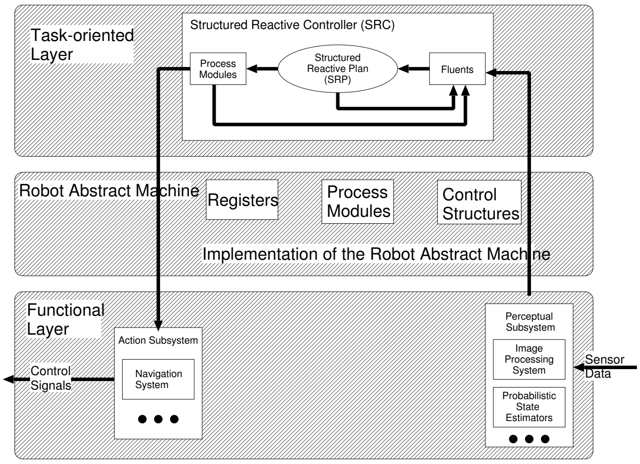

Fig. 2. The computational structure of the household robot control system. The figure depicts the two layers of the system. We consider the functional layer as being decomposed into a perception and an action subsystem. The task-oriented layer is constituted by the Structured Reactive Controller. The perception subsystem computes the information for setting the fluents of the Structured Reactive Controller. The process modules that are activated and deactivated by the Structured Reactive Plan control the modules of the action subsystem.

The task-oriented layer is constituted by the Structured Reactive Controller [2], [3] that specifies how the functions provided by the functional layer are to be coordinated in order to perform specific jobs such as the such as setting the table. The perception subsystem computes the information for setting the fluents of the Structured Reactive Controller. The process modules that are activated and deactivated by the Structured Reactive Plan control the modules of the action subsystem.

Structured reactive controllers work as follows. When given a set of requests, the structured reactive controller retrieves routine plans for individual requests and executes the plans concurrently. These routine plans are general and flexible — they work for standard situations and when executed concurrently with other routine plans. Routine plans can cope well with partly unknown and changing environments, run concurrently, handle interrupts, and control robots without assistance over extended periods. For standard situations, the execution of these routine plans causes the robot to exhibit an appropriate behavior in achieving their purpose. While it executes routine plans, the robot controller also tries to determine whether its routines might interfere with each other and watches out for non-standard situations. If it encounters a non-standard situation it will try to anticipate and forestall behavior flaws by predicting how its routine plans might work in the non-standard situation and, if necessary, revising its routines to make them robust for this kind of situation. Finally, it integrates the proposed revisions smoothly into its ongoing course of actions.

C. Simulation and Visualization

We have developed simulation tools for our kitchen and robot acting as robotic assistant using the Gazebo toolbox for 3D, physics-based robot simulation. The simulator is realistic along many dimensions. In particular, the simulator will use sensing and actuation models that are learned from the real robots. Models of rooms and their furnishing can also be acquired automatically using the methods for the acquisition of environment models, described in Section IV-A.

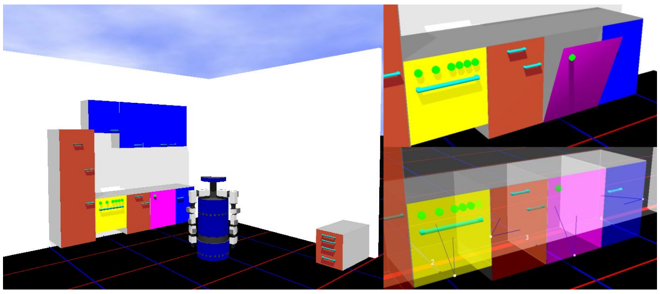

Fig. 3. The Assistive Kitchen and its simulation in Gazebo.

The use of these simulation tools promotes the research in assistive kitchen technology in various ways. First, the simulator supports generalization: we can model all kinds of robots in our simulation framework. We will also have different kitchen setups, which requires us to develop control programs that can specialize themselves for different robots and environments. The simulator allows us to run experiments fast and with little efforts and under controllable context settings. This supports the performance of extensive empirical studies.

III. Demonstration Scenarios

The demonstration scenarios are organized along two dimensions (see Table I). The first dimension is the domain tasks and activities under investigation. The second one is the research themes related to the cognitive aspects of perception, interpretation, learning, planning, and execution of these activities.

We start by looking at three scenario tasks in the context of household chores: setting the table, cooking, and performing household chores for an extended period. These tasks challenge cognitive systems along different dimensions, which we will discuss in the Section IV.

A. Scenario: Table setting

Table setting refers to the arrangement of tableware, cutlery, and glasses on the table for eating. Table setting is a

research theme	domain task/activity: table setting	cooking	household chore
perception	*	*	*
activity model	*	*	*
env. model	*	*	
motion primitives	*	*	
macro actions	*	*	
everyday activity			*

TABLE I
DIMENSIONS OF DEMONSTRATION SCENARIOS.

complex transportation task that can be performed as a single thread of activity.

There are various aspects of table setting that make it to a suitable challenge for cognitive systems: (1) the common-sense knowledge and reasoning needed are to perform the task competently; (2) the task itself is typically specified incompletely and in a fuzzy manner; and (3) the task can be carried out and optimized in many ways.

To account for incomplete task specifications, agents must use their experience or make an inquiry to determine who sits where and whether people prefer particular plates, cups, etc. Other missing information can be found in the world wide web such as which kinds of plates and cutlery are needed and where they should be placed.

Aspects to be considered for the optimization of the activity include the the position where the robot should stand to place the objects on the table, the decision if objects should be stacked or carried individually, etc. Also, the task requires the execution of macro actions such as approaching the cupboard *in order to* open it and get out the plates. Agents also have to consider subtle issues in action selection, such as whether or not to interrupt the carrying action in order to close the door immediately or later. Finally, actions require substantial dexterity. Putting objects on table where people are seated requires the robot to perform socially acceptable reaching actions.

B. Scenario: Cooking

Cooking is the activity of preparing food to eat. Unlike setting the table, cooking requires the selection, measurement and combination of ingredients in an ordered procedure. The performance of a cooking activity is measured in terms of the quality and taste of the meal and its timely provision.

Cooking involves the transformation of food, the control of physical and chemical processes, the concurrent execution of different activities, the use of tools and ingredients, and monitoring actions. Concurrent activities have to be timed such that all parts of the meal are done at the desired time. Cooking comprises many different methods, tools and combinations of ingredients — making it a difficult skill to acquire.

C. Scenario: Prolonged House Keeping

Prolonged house keeping and household chores are specific activities related to or used in the running of a household. The activities include cooking, setting the table, washing the dishes, cleaning, and many other tasks.

Performing household chores illustrates well the flexibility and reliability of human everyday activity. An important reason is the flexible management of multiple, diverse jobs. Humans as well as intelligent agents usually have several tasks and activities to perform simultaneously. They clean the living room while the food sits on the oven. At any time, interrupts such as telephone calls, new errands and revisions of old ones might come up. In addition, they have regular daily and weekly errands, like cleaning the windows.

Since doing the household chores is a daily activity and done over and over again, agents are confronted with the same kinds of situations many times, they learn and use reliable and efficient routines for carrying out their daily jobs. Performing their daily activities usually does not require a lot of thought because routines are general. This generality is necessary since jobs come in many variations and require agents to cope with whatever situations they encounter. The regularity of everyday activity also requires agents to handle interruptions that occur while carrying out their routines. Routines, like cleaning the kitchen, are often interrupted if a more urgent task is to be performed, and continued later on. Therefore, agents have to remember where they put things in order to find them again later.

A main factor that contributes to the efficiency of everyday activity is the flexibility with which different routines can be interleaved. Agents often work on more than one job at a time. Or, they do jobs together — operating the washing machine in, and getting stored food from, the basement.

People can accomplish their household chores well — even in situations they have not encountered yet — for two reasons: they have general routines that work well in standard situations and they are, at the same time, able to recognize non-standard situations and adapt their routines to the specific situations they encounter. Making the appropriate adaptations often requires people to predict how their routines would work in non-standard situations and why they might fail.

IV. Research Themes

The scenarios described in the previous section challenge cognitive systems along different dimensions. In this section, we discuss the research themes related to the cognitive capabilities needed to perform the tasks in the scenarios.

A. Acquisition and Use of Environment Models

Maps or environment models are resources that the Assistive Kitchen uses in order to accomplish its tasks more reliably and efficiently. Acquiring a model of an assistive kitchen is very different from environment mapping done by autonomous robots. For mapping the Assistive Kitchen the robot can make use of the sensors of the environment. Semantic information is also associated with RFID tags in the environment and the sensors are services providing information about themselves and their use. In contrast to many other robot mapping tasks where the purpose of mapping is to support navigation tasks, the kitchen map is a resource for understanding and carrying out household chores. To this end, the map needs to have a richer structure, explicitly reference task relevant objects such as appliances, and know about the concept of containers and doors, such as cabinets and drawers.

Here, we consider mapping to be the following task: *Given* (1) a sensor-equipped kitchen where appliances and other pieces of furniture might or might not be tagged with RFID tags that have information associated with them (such as their size) and (2) a stream of observations of activities in this kitchen acquired by the various sensors *acquire* a semantic, 3D object map of the kitchen.

We have developed a system for the acquisition of object maps (see Fig. 4) based on a set of comprehensive geometrical reasoning techniques that extract semantic object information from 3D point cloud data [11]. The object map comprises the static parts of the environment such as the kitchen furniture, appliances, tables, and walls, saved in OWL-DL format in our knowledge base, which is shared between different applications and robots, and can be queried to retrieve higher level information. To build accurate and robust robot plans, we make use of the object map by automatically importing it in our 3D simulation environment.

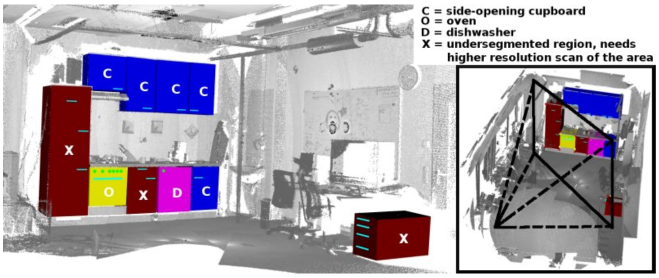

Fig. 4. An Object Map of the Assistive Kitchen.

The maps also have to contain sensors of the environment and their locations in order to semantically interpret the sensor data. For example, the mapper has to estimate the position and orientation of a laser range sensor using its own motions and their effects on the sensor data, in order to infer that the sensor can be used for determining its own position and how. Or, the robot has to locate an RFID tag reader, for example using the estimated position of its gripper and observing which of the sensors reports the RFID identifier of its gripper. Knowing that its gripper is inside a cabinet it can infer that the respective sensor can be used to sense what is inside this cabinet.

Further information can be acquired by observing activities in the kitchen. By describing objects and places as action-related concepts, i.e. by their roles in actions, the system can infer their function and refer to them not based on the appearance or position but on the purpose they serve for. Figure 5 shows an example of a place that has been learned as the one where the robot stands when picking up cups from the cupboard. Action-related concepts are in many cases a very natural representation since most objects in a household are designed for a certain purpose.

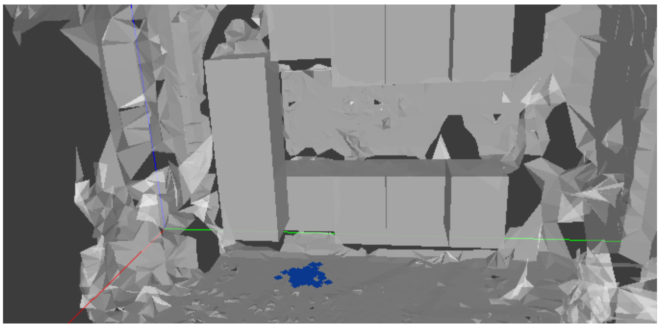

Fig. 5. Positions learned as "pick-up cup place" from observations of table-setting actions.

Finally, the geometry of the environment and recognized activities are used to fine-tune the sensors for particular activity recognition tasks.

B. Acquisition and Use of Activity Models

The first research theme is concerned with the observation of human activity in the kitchen in order to acquire abstract models of the activity. The models are then to be used to

- answer queries about the activity,
- monitor activity and assess its execution,
- generate visual summaries of a cooking activity including symbolic descriptions, and
- detect exceptional situations and the need for help.

In order to do so, the system is required to answer questions including the following ones: What are meaningful sub-activities of "setting the table" and why? Which eating utensils and plates have to be used for spaghetti and how should they be arranged? Does John prefer a particular cup? Where do people prepare meals? How do adults set the table as opposed to kids? And why? Why does one put down the plates before bringing them to the table? (To close the door of the cupboard.) What happens after the table is set? Where are the forks kept? What is the fork used for?

The models needed for answering these questions are to be acquired (semi-)automatically from the world wide web, from observing people, and by asking for advice.

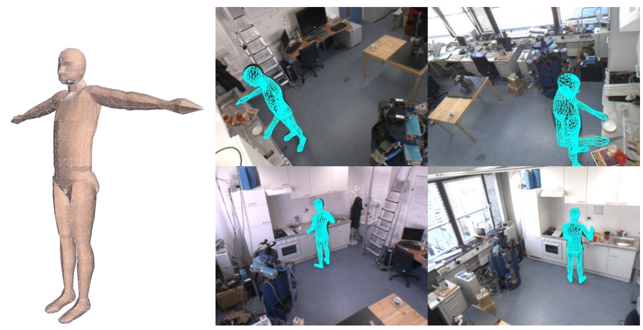

Fig. 6. The model used for human motion capture (left), and the views of the four cameras in the kitchen (right).

One of the technologies we use to observe and interpret human activity is the markerless human motion capture system we have developed [1]. This system enables us to retrieve accurate 3D human posture of people performing manipulation tasks in the kitchen, as depicted in Fig. 6. The algorithm uses an anthropometric human model to track persons based on an initialization and observation data from the four ceiling-mounted CCD cameras. The model can be adapted to accurately capture the appearance of a broad range of humans. As output we get the articulated joint angles of the human skeleton, that can further be used for action classification or interpretation of human intentions. We are also investigating ways to incorporate complementary sensor data to ease the computationally expensive task so that we can improve the motion capture algorithm towards real-time processing.

Furthermore, we use the gaze-aligned head mounted camera and small inertial measurement units, depicted in Fig. 7, to record the person's gaze and arm posture in everyday manipulation tasks. Fig. 8 depicts an image from the head mounted camera, and some of the intermediate results in extracting a 3D hand and object model from them.

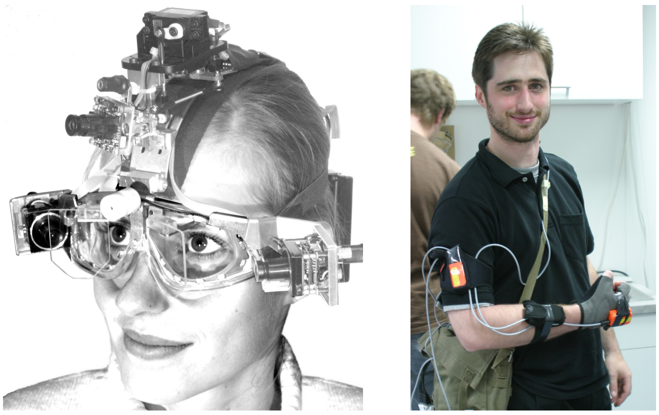

Fig. 7. The gaze-aligned head mounted camera (left) and inertial measurement units (right).

C. Action and Motion Primitives

People doing their household activities perform very complex movements smoothly and effortlessly, and improve such movements with repetition and experience. Such activities like putting the dishes in the cupboard, or setting the table, are simple for people but present challenges for robots.

Consider picking up a cup from the table and placing it in the cupboard. Any person will turn and walk towards the cup, while simultaneously stretching the arm and opening the hand. Then the hand will grasp the cup firmly, and lift it. It will apply just the right amount of force, because through experience, the person has learned how much a cup should weigh. If the cup were heavier as expected, this would be detected and corrected immediately. Then, taking the cup close the body for a better stability, the person will walk to the cupboard, open the door with the other hand, and place the cup inside.

Several noteworthy things take place. First, the motions used to get the cup are similar to the ones used for reaching and picking up other objects. Such movements can be formed by combining one or more basic movements, called motion primitives. These primitives can be learned by observation

Fig. 8. Processing steps in fitting object and hand models to the images from the head mounted camera: 1) matched object model; 2) matched 2D hand model; 3) matched 3D hand model.

and experimentation. Each one of this primitives can be parameterized to generate different movements. Second, it is known that noise and lag are present in the nervous system, which affect both sensing and motor control. But people manage despite this limitations to have elegant and precise control of our limbs. A robot with a traditional controller would have great difficulties carrying out simple movements in such conditions. Humans deal with this problem by building forward and inverse models for motor control [12]. Third, during a movement, any abnormal situation is quickly detected and a corrective action is taken.

This research theme is concerned with endowing a robot with similar capabilities. The robot will observe the activities of people, obtain motion primitives from them, and improve them through own experimentation. Models of activity are also learned for reaching and grasping movements, and include information about the effects of the control commands. This allows the robot make predictions and detect abnormal situations when the sensory data differs from the expected values.

D. Planning and Learning Macro Actions

The next higher level of activities in the kitchen are macro actions. We consider macro actions to be the synchronized execution of a set of action primitives that taken together perform a frequent macro activity in the task domain. These activities are so frequent that the agents learn high-performance skills from experience for their reliable and flexible execution. Examples for such macro activities are opening a tetrapak and filling a cup with milk, opening a cabinet to take a glass, or buttering a bagel.

Here the challenge is to compose macro actions from coordinated action primitives such that the resulting behavior is skillful, flexible, reliable, and fluent without noticeable transiting between the subactions [14]. Another challenge is that such macro actions must be learned from very little experience — compared with other robot learning tasks [6].

The learning of macro actions is an application of *action aware control* [13] where the agents learn performance and predictive models of actions and use these models for planning, learning, and executing macro actions. The composition of such macro actions requires the application of transformational learning and planning methods and the combination of symbolic and motion planning with learned dynamic models.

E. Self-adapting High-level Controller

Robotic agents can not be fully programmed for every application. Thus, in this research theme we realize robot control programs that specialize to their respective robot platform, work space, and tasks (see Fig. 9).

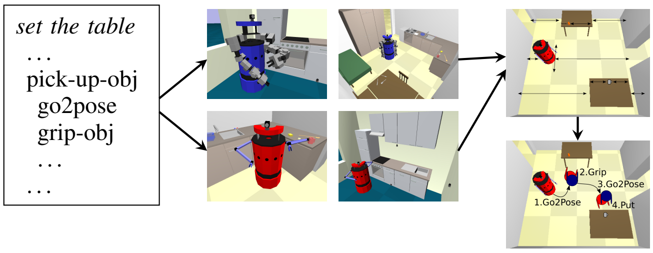

Fig. 9. Self-adaptation of different robots in different kitchens.

We realize a high-level control program for the specific task of setting the table. The program learns from experience where to stand when taking a glass out of the cupboard, how to best grasp particular kitchen utensils, where to look for particular cutlery, etc. This requires the control system to know the parameters of control routines and to have models of how the parameters change the behavior [4]. Also, the robots perform their tasks over extended periods of time, which requires very robust control [5].

F. Learning to Carry out Abstract Instructions

The research theme discussed in this paper is the acquisition of new high-level skills. Let us consider cooking pasta as an illustrative example. Upon receiving "cook pasta" the robot retrieves instructions from webpages such as *ehow.com*. These instructions are typically sequences of steps to be executed in order to carry out the activities successfully. The challenges of this execution research theme are: (1) translate the abstract instructions into an executable robot control program, (2) disambiguate the often incomplete task descriptions given in natural language and supplement missing information through observations of kitchen activities, (3) transform the action sequence into an activity structure that can be carried out more reliably, efficiently, and flexibly. Instructions typically abstract away from these aspects of activity specification.

The procedure for transforming natural language task instructions into executable robot plans comprises the following steps (Fig. 10): A syntax parser analyzes the sentence structure and labels the parts of speech. The resulting syntax tree is then transformed to instructions including the action to be performed, the object(s) to be manipulated, possible pre- and postconditions as well as parameters like a desired location. Wordnet and Cyc are used for determining the meaning of the words. An intermediate plan representation is created in the knowledge base which can be exported as an executable robot plan.

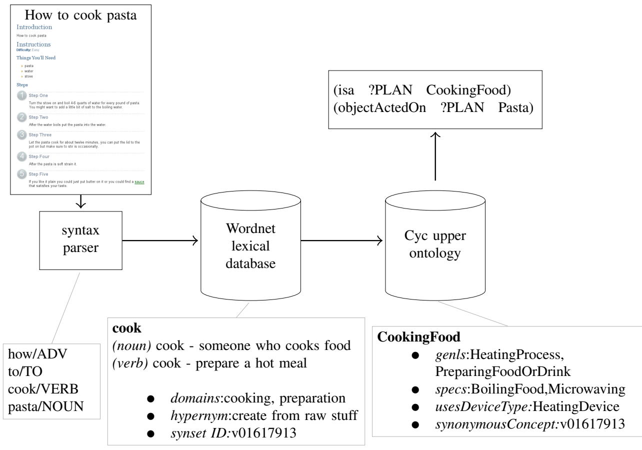

Fig. 10. Import procedure of natural language task descriptions into executable robot plans.

We have successfully imported (almost) working plans from descriptions from ehow.com using the described procedure, but several hard challenges remain in this scenario. A general, robust solution for translating abstract instructions into working robot control programs requires answers to the following research questions.

(1) How can ambiguities in the descriptions be resolved and which missing parameters have to be inferred? For example, a description for setting a table does not explicitly state that the objects have to be placed on the table but only describes the positions relative to each other.

(2) How can the plan libraries of autonomous household robots be specified so generally, reliably, transparently, and modularly that a robot can compose working plans from abstract instructions? In order for newly composed sequences of plan steps to work it helps if the individual plan steps are specified as "universal plans", that is they achieve – if necessary – all preconditions needed for producing the desired effects.

(3) Debugging newly created plans from instructions requires the robot to predict what will happen if it executes the new plan, to identify the flaws of the plan with respect to its desired behavior, and to revise the plan in order to avoid the predicted flaws.

(4) Optimizing tasks like table setting also requires the technical cognitive system to observe people setting the table, to infer the structure of the activity and reason about why people perform the task the way they do instead of following the abstract instructions. This way the robot would learn that people stack plates when carrying them in order to minimize the distance they have to walk. The robot

would then transform its plan analogously and test whether this change of activity structure would result in improved performance.

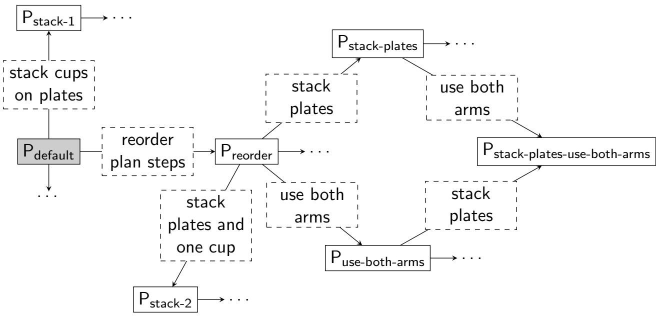

Fig. 11. Plan transformations performed by the TRANER planning system in order to optimize a table setting activity.

Figure 11 sketches the search for optimized table setting strategies performed by our planning system. The planning system called TRANER transforms a default plan for table setting (in which the robot is asked to carry objects to the table one by one) into an optimized plan (in which the robot stacks plates, takes cups in both hands, leaves cupboard doors open while setting the table and position itself to reach multiple objects and positions from the same position). The unique feature of TRANER [9] is that it applies very general plan transformations such as improve efficiency of transport tasks by stacking objects, or use both hands to concurrent reactive plans.

V. CONCLUDING REMARKS

This paper has presented assistive kitchens as demonstration platforms for cognitive technical systems that include various research challenges for cognitive systems. In particular, we expect the investigation of cognitive capabilities in the context of human everyday activity, which has received surprisingly little attention in previous research efforts, to substantially promote the state-of-the-art of cognition for technical systems.

REFERENCES

[1] J. Bandouch, F. Engstler, and M. Beetz. Accurate human motion capture using an ergonomics-based anthropometric human model. In *Proceedings of the Fifth International Conference on Articulated Motion and Deformable Objects (AMDO)*, 2008. Accepted for publication.

[2] M. Beetz. *Concurrent Reactive Plans: Anticipating and Forestalling Execution Failures*, volume LNAI 1772 of *Lecture Notes in Artificial Intelligence*. Springer Publishers, 2000.

[3] M. Beetz. *Plan-based Control of Robotic Agents*, volume LNAI 2554 of *Lecture Notes in Artificial Intelligence*. Springer Publishers, 2002.

[4] M. Beetz. Plan representation for robotic agents. In *Proceedings of the Sixth International Conference on AI Planning and Scheduling*, pages 223–232, Menlo Park, CA, 2002. AAAI Press.

[5] M. Beetz and H. Grosskreutz. Probabilistic hybrid action models for predicting concurrent percept-driven robot behavior. *Journal of Artificial Intelligence Research*, 24:799–849, 2005.

[6] M. Beetz, A. Kirsch, and A. Müller. RPL-LEARN: Extending an autonomous robot control language to perform experience-based learning. In *3rd International Joint Conference on Autonomous Agents & Multi Agent Systems (AAMAS)*, 2004.

[7] R. Brachman. Systems that know what they're doing. *IEEE Intelligent Systems*, pages 67 – 71, November/December 2002.

[8] M. Buss, M. Beetz, and D. Wollherr. CoTeSys — cognition for technical systems. In *Proceedings of the 4th COE Workshop on Human Adaptive Mechatronics (HAM)*, 2007.

[9] A. Müller, A. Kirsch, and M. Beetz. Transformational planning for everyday activity. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS'07)*, pages 248–255, Providence, USA, September 2007.

[10] R. B. Rusu, B. Gerkey, and M. Beetz. Robots in the kitchen: Exploiting ubiquitous sensing and actuation. *Robotics and Autonomous Systems Journal (Special Issue on Network Robot Systems)*, 2008.

[11] R. B. Rusu, Z. C. Marton, N. Blodow, M. E. Dolha, and M. Beetz. Functional Object Mapping of Kitchen Environments. In *Accepted for publication in Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, September 22-26*, 2008.

[12] S. Schaal and N. Schweighofer. Computational motor control in humans and robots. *Current Opinion in Neurobiology*, 15:675–682, 2005.

[13] F. Stulp and M. Beetz. Action awareness – enabling agents to optimize, transform, and coordinate plans. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2006.

[14] F. Stulp and M. Beetz. Refining the execution of abstract actions with learned action models. *Journal of Artificial Intelligence Research (JAIR)*, 32, June 2008.