SEVENTH FRAMEWORK PROGRAMME

Project Acronym:	Xperience
Project Type:	IP
Project Title:	Robots Bootstrapped through Learning from Experience
Contract Number:	270273
Starting Date:	01-01-2011
Ending Date:	31-12-2015

XPERIENCE.ORG

Deliverable Number:	D5.2.3
Deliverable Title:	Third Demonstration on Scenario 1: Execution of complex plans
Type (Internal, Restricted, Public):	PU
Authors:	Norbert Krüger, Tamim Asfour, Aleš Ude, Dirk Kraft, Mirko Wächter, Martin Do, Andrej Gams, Bojan Nemec, David Schiebener and Rüdiger Dillmann
Contributing Partners:	SDU, IIT, KIT, JSI, UGOE

Contractual Date of Delivery to the EC: 31-01-2013
Actual Date of Delivery to the EC: 07-04-2013

Contents

Chapter 1

Executive Summary

This deliverable deals with the demos in the context of WP5.2 In Chapter 2 we show structural bootstrapping first within the objective of learning to wipe and secondly in a transfer of the information learned in a wiping situation to stirring. Chapter 3 discusses a demo that shows a longer action sequence to cleanup a table. In the remaining part we describe our demonstrations within the cutting domain – transfer between knifes, an analysis of human cutting behaviour and cutting execution in a bigger scenario with the help of semantic event chains – and our progress on combining vision and physical interaction to learn object models. The execution of these demos is shown on different platforms at KIT, JSI, IIT, SDU and UGOE.

Chapter 2

Main Demo 1: Wiping, Stirring and Mixing

This demonstration covers the learning of wiping from human observation as well as the bootstrapping of stirring actions through wiping actions. The two demonstrations below are major blocks for the demonstration of the proof of concept example.

2.1 Learning How to Wipe: A Case Study of Structural Bootstrapping from Sensorimotor Experience

To demonstrate the validity of the Xperience learning cycle, a behaviour for the learning of a wiping action has been implemented which allows the continuous creation, evaluation, and refinement of models and representation extracted from sensorimotor experience gained through observation and exploration. The learning process is triggered by the observation of a human wiping demonstrations which are represented in the form of a generalized movement primitives. Using these action representations, the robot conducts a wiping experiment with various objects which can be found in the scene. To determine the properties of the objects such as softness, size, and appearance, the robot explores each object visually and haptically. Based on the execution of the wiping action and its observation, action parameters are specified in order to maximize the wiping effect considering the specific object in hand and the current scene. Based on this sensorimotor experience internal models are learned which represent the object-action-effect relations. In subsequent iterations of the learning cycle, the learned models are used to identify potential wiping objects in order to guide the exploration and the learning behaviour. To increase the efficiency within each cycle, action parameters and their effects are predicted using theses models. The cycle has been implemented on the humanoid platform ARMAR-IIIb and experiments with various objects have been conducted (see [2] and **Deliverable D3.1.2**). The results of this work are shown in the attached video **WipingLearningCycle.avi**.

2.2 Transfer of Stirring to Mixing at the Trajectory Level

The video **WipingToStirring.mp4** demonstrates how bootstrapping the learning process with a similar activity speeds up sensorimotor learning. Our experimental setup for learning of stirring behaviour is shown in Figure 2.1. It is composed of two KUKA LWR robots, equipped with a Barrett hand and a two finger gripper. The task is to learn how to stir in a metal pad of diameter of 21 cm using a wooden spoon. The first part of the video shows learning of stirring without prior knowledge of the stirring trajectory. The criterion function for motion learning uses the desired force $\mathbf{F}_d$ with which the robot should move along the edge of the pot. For learning we used the Repetitive Control(RC) algorithm applied to periodic DMP representing the movement (see Deliverable **D3.1.2**, Section 2.6 for details). As we can see in the video, the RC algorithm requires from 15 to 25 repetitions to obtain a satisfactory stirring trajectory. Next case considers learning where the adaptation process is bootstrapped with wiping trajectory, where the information about the similarity between stirring and wiping is extracted from the semantic event chains of the two actions. The wiping trajectory is obtained by imitation learning using kinaesthetic guiding, as shown in the continuation of the video. The robot learned the policy in approximately 7 cycles with prior knowledge taken from wiping motion. Time evolution of learned trajectories are shown also in Figure 2.2, where we can clearly see the improved learning speed using prior knowledge.

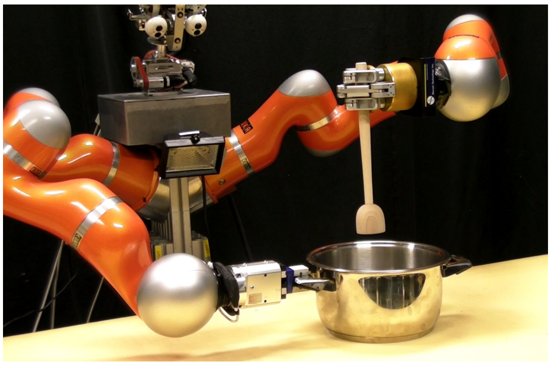

Figure 2.1: Experimental setup for stirring learning

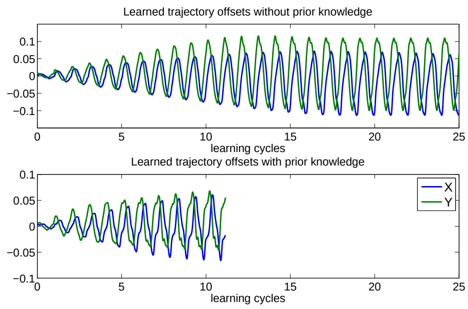

Figure 2.2: Time evolution of stirring trajectories in x-y plane

Chapter 3

Main Demo 2: Cleaning a Table on iCub Robot in Yarp Architecture

In [1] we investigated the implementation of the demonstrator using the behaviour based architecture introduced in Year 2. The goal of the demonstrator is that of showing the "clean the table" task. The architecture has been now fully extended through a plug-in system included in the YARP middleware.

The description of the architecture is reported in D1.2.1 (report on the blueprint of the cognitive architecture). In short, a system of plugins loaded at the level of YARP ports implements a behaviour based system similar to the subsumption architecture. This represents only the first level of the complete Xperience architecture and could be connected to the higher-level planners/reasoning systems. The plugins implement a "port monitor" that is a Lua script that enables various logic (FSM) to be executed upon reception of a message at the port input. The special port monitor implemented at this point arbitrates (port arbitrator) the incoming messages according to a priority rule which in fact implements the subsumption architecture.

To demonstrate the applicability and advantages of our approach, we present an experiment with the iCub humanoids robot. The goal of the task, as shown in the activity diagram, is to clean the table by removing all objects and place them in a bucket located alongside the table. This experiment is completely built using modules from the iCub software repository. The experiment focuses on reusing (with no modifications) existing modules and by extending the required functionalities using port plug-ins. The overall behaviour of the "clean the table" is shown by the simplified activity diagram in Figure 3.1.

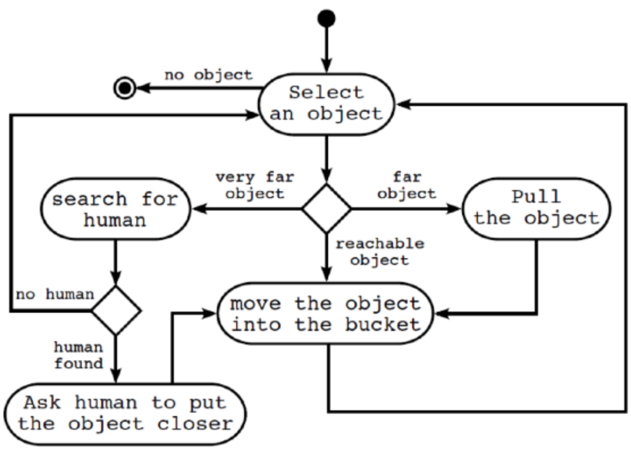

Figure 3.1: The simplified activity diagram that illustrates the table-cleaning.

We allowed the robot to use a tool at his disposal (a rake), located on a rack, to reach objects of interest that are out of its workspace. The modules that allow the robot to grasp and use the tool are implemented as described in [4]. Furthermore, we consider also the case in which the object is so far that it cannot be reached even by the use of the tool. In this case the robot should look for a human and asks her intervention (put the object within reach).

Figure 3.2 shows the experimental setup and it illustrates the three areas in which objects can be placed. The activity diagram depicted in Figure 3.1 may give the impression that the task is only composed of a few simple steps that the robot should follow to accomplish it. But in fact, there are many uncertainties and unexpected conditions which should be taken into consideration to make the task robust. For example, the proper decision should be taken if an object drops from the hand while the robot is placing it into the bucket. Similarly the robot should behave appropriately while it is holding the tool to pull the object closer, the human might intentionally intervene and move the object within the iCub's workspace. Considering all possible uncertainties, in fact, reveals the underlying complexity of the task which requires that many modules (e.g, for perception, action and coordination) are properly used and orchestrated (e.g, coordinating robots, gaze, arm, speech) to perform the required task.

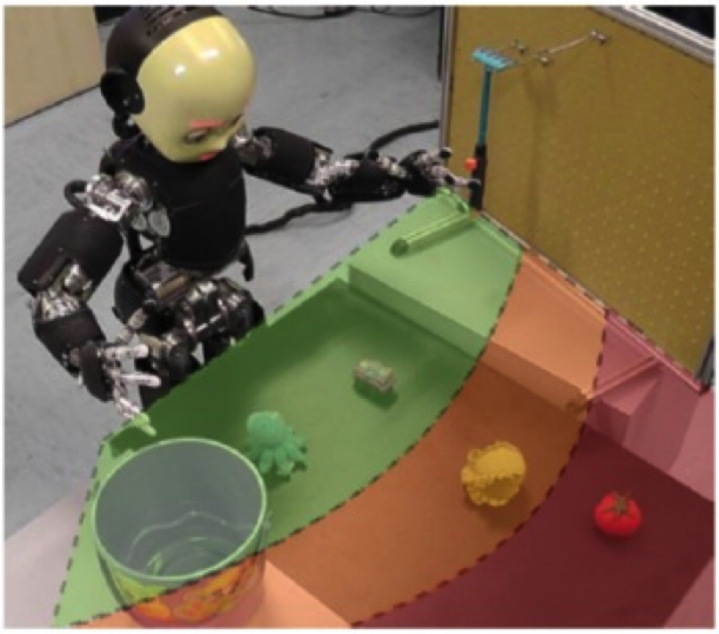

Figure 3.2: The experimental setup of table-cleaning application. The reachable zone is depicted in green, the orange zone represents the zone reachable with the tool and finally the red zone indicates the unreachable space, for which the robot needs human intervention.

The diagram in Figure 3.3 shows the module connections (and actual modules in the iCub repository) and the following images in Figure 3.4 show the iCub during the cleaning task. The video **iCubCleaningTheTable.mp4** shows the complete experiment (4 minutes in length).

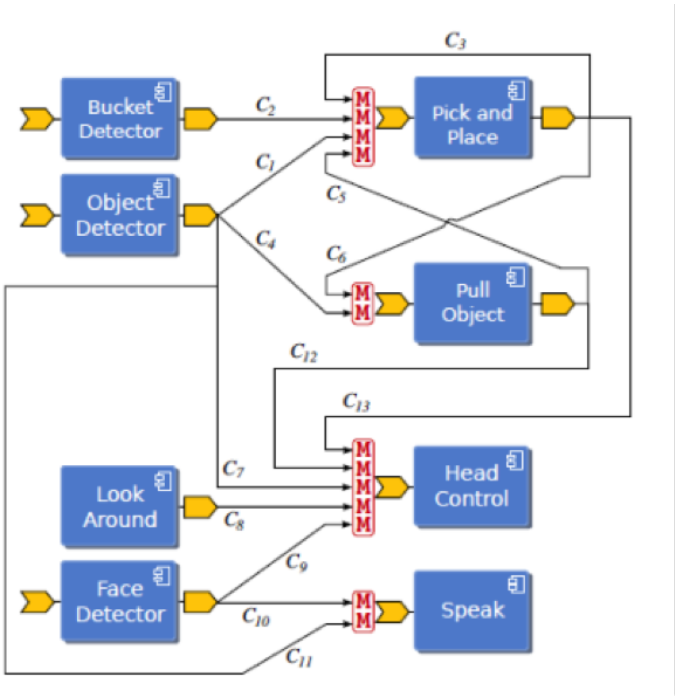

Figure 3.3: Configuration of the modules for table-cleaning application.

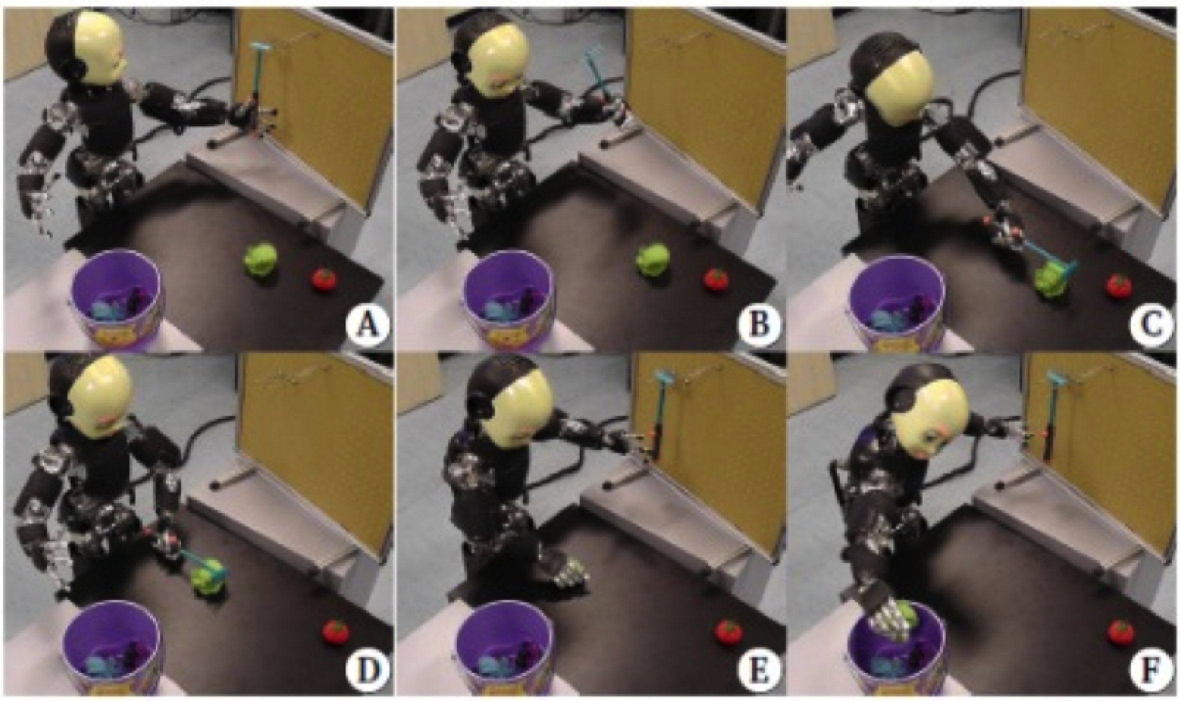

Figure 3.4: The iCub performing table-cleaning using a tool (rake). The robot take the tool (A), reaches for the object (B,C), pulls the object (D), grasps the object (E) and finally places it into the bucket (F).

Chapter 4

Additional Demonstrations

Besides the main demonstrations detailed in the two previous chapters we have implemented additional demos to explore specific aspects. In Section 4.1 we present our work related to cutting food objects by robots and humans. While in Section 4.2 we detail the work on combining vision and physical interaction for learning rigid objects.

4.1 Human and Robot Cutting

We gave addressed different aspects of the problem of cutting food related objects on robot platforms at partner SDU and UGOE. Our aim is to integrate the different components related to cutting on one platform in the next year. In Section 4.1.1 we show a system that is able to grasp unknown knifes and use them to cut. Our initial analysis of human cutting behaviour and a corresponding setup we build is described in Section 4.1.2. While Section 4.1.3 shows a video that presents the use of semantic event chains for longer actions sequences also containing cutting actions.

4.1.1 Leaning Where to Grasp Unknown Knifes for Cutting

We developed a demonstrator that allows for the transfer of grasp for cutting to unknown knifes. The video **GraspingForCutting.mp4** shows how we use our simulation setup to generate experiences of grasping a knife and using it for cutting, these are used to learn which visual constellations predict grasp positions that lead to successful grasp-cut action pairs. We implemented these methods on the MARVIN platform at SDU where they enable us to grasp previously unseen knifes. For that a RGB-D camera is used to get a representation oft the current scene. The video shows how the learned model is used to predict good grasp positions and how once the knife is grasped, a cutting trajectory based on human demonstration is used to cut the fruit.

4.1.2 Analysis of Human Cutting Behaviour

To more accurately study trajectories from human cutting we developed a setup at SDU with a small 6D F/T sensor, a 6D magnetic pose tracker and printed parts that allow the mounting of the blade. In our video **CutAnalysis.mp4** we show how this setup is used to log data from demonstrated cutting trajectories of several different fruits and vegetables and show how these trajectories vary between different human demonstrations. This provides data for formalising the space in which successful cutting is possible and in which the robot later will compute successful cutting trajectories.

4.1.3 Cutting with the help of SECs

We have created a library of manipulation actions based on the semantic event chain (SEC) framework. The actions are defined by a SEC matrix that shows how the relations between objects change throughout an action. In the video **CuttingSEC.mp4**, a long kitchen scenario is demonstrated, which is a sequence of actions from our library. The scenario is as follows:

- Pick a zucchini and place it on a cutting board.
- Grasp a knife, cut the zucchini into pieces and release the knife on the table.
- Pick up the board and unload the zucchini pieces into a bowl.
- Grasp a spoon, stir into the bowl and release the spoon on the table.

The whole sequence relies on detecting the touching relations between different objects, which are detected by different sensors (here we use force, position and tactile sensors). Our vision system detects the position and orientation of the objects. This information is used to guide the robot toward the correct positions and orientations.

4.2 Combining Vision and Physical Interaction for Learning of Textured and Nontextured Rigid Objects

We have extended our work on combining visual perception and physical interaction for autonomous segmentation and learning of unknown objects [3]. In the last period, we have reported on how a robot can support visual object segmentation by actively pushing and thereby moving an object. This allows reliable segmentation of unknown objects even in extremely cluttered environments. We demonstrated that the obtained segmentation can be sued to learn a visual object descriptor, and to roughly estimate the object extent in order to bootstrap a reactive grasping attempt. Using the segmentation to estimate a promising initial grasping pose, the grasp reactively corrected during its execution based on feedback from haptic and force sensors.

Figure 4.1: Examples for interactive object segmentation in extreme clutter

The interactive object segmentation approach was, however, limited to textured objects, as SIFT features were the main means for estimation the object motion after it has been pushed. In this period, we were able to improve our approach by devising a new algorithm for object relocalization that is based on colour-annotated 3D point clouds (RGB-D) and can seamlessly handle textured as well as nontextured objects. We are thus now able to apply our approach to a large variety of rigid objects. For more details, see **D2.1.2** Chapter 3.

The attached video **InteractiveSegmentationOfRigidObjectsInClutter.mpg** shows the interactive segmentation of an unknown object in an extremely cluttered scene by the robot ARMAR-III.

References

[1] Paikan A., Tikhanoff V., Metta G., and Natale L. Enhancing software module reusability using port plug-ins: an experiment with the icub robot. In *Submitted to IROS*, 2014.

[2] M. Do, J. Schill, J. Ernesti, and T. Asfour. Learn to wipe: A case study of structural bootstrapping from sensorimotor experience. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[3] D. Schiebener, A. Ude, and T. Asfour. Physical interaction for segmentation of unknown textured and non-textured rigid objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014.

[4] V. Tikhanoff, U. Pattacini, L. Natale, and G. Metta. Exploring affordances and tool use on the icub. In *IEEE/RAS International Conference of Humanoids Robotics (Humanoids)*, Atlanta, GA, USA, 2013.