



FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D7.2005

Refined Specification of Key Experiments and Implementation Status

Due date of deliverable: 31/12/2005

Actual submission date: 25/01/2006

Start date of project: January 1st, 2004

Duration: 48 months

Contributing partners:

IPA, LAAS, UniKarl, UvA, EPFL, UH, UniBi

Revision: final

Dissemination Level: PU

Executive Summary

This joint deliverable reports on the progress made in Research Area 7 *System Level Integration and Evaluation* (RA7) during phase 2 (Jan-Dec 2005) of Cogniron.

- Scientific and technical advances in each of the three Key Experiments that were defined in [2] are presented in sections 2, 3, and 4. This progress is based on more detailed specifications than before, they are introduced in section 1.3.5 (p.10) and presented in appendix E (p.84).
- Given the increased amount of integration, fostering technology transfer between project partners has grown in importance. We answered this challenge by creating a collaborative exchange platform, presented in section 5 (p.31).

Role of the Joint Deliverable D7.2 in Cogniron

According to the project's description of work [1], activity in RA7 addresses the specification, demonstration and validation of Cogniron's research issues in terms of Key Experiments. A Key Experiment focuses on one or more fundamental abilities of a cognitive robot system. The various demonstrations will serve as integration platforms for Research and Technology Development (RTD) performed throughout the project and will **reflect the increments of the achieved results** towards a cognitive robot companion.

This context places RA7 activities at multiple crossroads. First and foremost, it aims at producing both *scientific* system-level integration as well as providing *demonstrator platforms* with concrete implementations on real robots in realistic settings. Cogniron is an Integrated Project, thus a considerable part of the associated scientific and technological transfers take place on a European level. This widens the scope of RA7 leadership, which has to strike a balance between technical and non-technical aspects, as presented in section 1.1 (p.6).

Contents

1	General Issues in RA7 System Level Integration and Evaluation	6
1.1	Scope of RA7	6
1.2	Addressing Integration and Evaluation	6
1.3	Progress in RA7 During Phase 2	8
1.3.1	Key Experiment 1 <i>Robot Home Tour</i>	8
1.3.2	Key Experiment 2 <i>The Curious Robot</i>	9
1.3.3	Key Experiment 3 <i>Learning Skills and Tasks</i>	9
1.3.4	Software Integration	10
1.3.5	Refined Specifications	10
1.4	Conclusion and Outlook	11
2	Key Experiment 1 <i>Robot Home Tour</i>	12
2.1	Role of KE1 in Cogniron	12
2.2	Status of KE1 at T0+24 (December 2005)	12
2.2.1	CF-PTA <i>Person Tracking and Detection of Attention</i>	12
2.2.2	CF-DLG <i>Multi Modal Dialogue</i>	15
2.2.3	CF-ROR <i>Resolving Multi Modal Object References</i>	15
2.2.4	CF-LOC <i>Dialogue or Perception Based Localization</i>	15
2.2.5	SF <i>Laser-Based Navigation</i>	15
2.3	Conclusion and Outlook	16
3	Key Experiment 2 <i>The Curious Robot</i>	17
3.1	Role of KE2 in Cogniron	17
3.2	Status of KE2 at T0+24 (December 2005)	18
3.2.1	CF-MHP <i>Manipulation in Human Presence</i>	18
3.2.2	CF-OR <i>Object Recognition</i>	18
3.2.3	CF-NHP <i>Navigation in Human Presence</i> and CF-SOC <i>Socially Acceptable Interaction with Regard to Space</i>	19
3.2.4	CF-TBP <i>Tracking of Human Body Parts for Observation</i>	19
3.2.5	CF-GR <i>Gesture Recognition</i>	19
3.2.6	CF-RET <i>Reasoning About Tasks and its Own Abilities</i>	20
3.2.7	Supporting Services	20
3.2.8	Status of KE2 Script <i>Robot Initiative and Intentionality</i>	21
3.2.9	Status of KE2 Script <i>Robot Curiosity</i>	22
3.3	Conclusion and Outlook	22
4	Key Experiment 3 <i>Learning Skills and Tasks</i>	23
4.1	Role of KE3 in Cogniron	23
4.2	Status of KE3 at T0+24 (December 2005)	23
4.2.1	CF-LIF <i>Learning Important Features of a Task</i>	24
4.2.2	CF-RG <i>Learning to Reproduce Gestures</i>	24
4.2.3	CF-LCT <i>Learning Complex Tasks</i>	25
4.2.4	CF-TBP <i>Tracking of Body Parts</i>	26
4.2.5	CF-ACT <i>Recognition of Human Activities</i>	26
4.2.6	CF-GR <i>Gesture Recognition</i>	26

4.2.7	CF-OR <i>Object Recognition</i>	27
4.2.8	Status of KE3 Script1: Arranging and interacting with objects	27
4.2.9	Status of KE3 Script 2: Learning to serve a guest	28
4.3	Conclusion and Outlook	28
5	Software Integration	31
5.1	Role of Software Integration in Cogniron	31
5.2	From Cogniron Functions to Software: Services and Data Structures	31
5.3	Enabling Software Exchange: Module Market	32
5.3.1	Modules	33
5.3.2	Tools and Libraries	33
5.3.3	Other Elements	33
5.4	Conclusion and Outlook	33
6	References	35
6.1	Project Documents and Resources	35
6.2	Applicable Documents	35
6.3	Reference Documents and Resources	37
A	Account of Technology Transfer Between UniBi and LAAS	38
A.1	KE1 Software Integration Context	38
A.2	Objectives of Technology Transfer	39
A.3	Preparation and On-Site Work	39
A.4	Results	39
A.5	Conclusion	42
B	Detailed Integration Account for KE2	43
B.1	Objectives and Organization	43
B.2	Experiment KE2-1: Object Modeling and Handling by a Mobile Manipulator	43
B.2.1	Objectives and Context	43
B.2.2	Robot Platform	44
B.2.3	Scenario	44
B.2.4	Software Architecture	45
B.2.5	Implemented Services	46
B.2.6	Results and Conclusions	51
B.3	Experiment KE2-2: Towards Human Friendly Navigation	52
B.3.1	Objectives and Context	52
B.3.2	Robot Platform	52
B.3.3	Scenario	52
B.3.4	Software Architecture	53
B.3.5	Implemented Services	53
B.3.6	Supervision for Human-Robot Interaction	59
B.3.7	Results and Conclusions	61
B.4	Experiment KE2-3: Curiosity and Learning	61
B.4.1	Objectives and Context	61
B.4.2	Robot Platform	62
B.4.3	Scenario	62

B.4.4	Software Architecture	63
B.4.5	Implementation	69
B.4.6	Results and Conclusion	72
B.5	Conclusions	72
C	Table of Cogniron Functions	73
C.1	Summary of Questionnaire Results	74
C.2	CF Implementation and Testing Status Overview	80
D	Table of Software Modules	81
D.1	Glossary	81
D.2	Software Available Through the Wiki	81
E	Key Experiment Specification Details	84
E.1	KE1 Details	84
E.2	KE2 Details	88
E.3	KE3 Details	89

1 General Issues in RA7 System Level Integration and Evaluation

This section gives an overview of the more detailed sections and appendices of the deliverable and presents some general issues that were addressed in RA7 during phase 2 of Cogniron. The scope of RA7 is presented in section 1.1 and its implications for work on integration and evaluation are presented in 1.2. Section 1.3 on integration progress starts with summaries of advancement in each KE (see 1.3.1, 1.3.2, and 1.3.3), followed by an introduction to software integration aspects in 1.3.4 and an account of refined specifications in 1.3.5. Conclusions and an outlook are found in section 1.4, followed by the more detailed parts of this deliverable.

1.1 Scope of RA7

As stated at the very beginning of this document under “Role of the Joint Deliverable D7.2 in Cogniron”, the goal of demonstrating scientific results on real machines places RA7 at multiple crossroads. Science, engineering, and leadership challenges intertwine, as sketched in figure 1.

It can be seen that two kinds of innovation are important in RA7. Scientific activity takes place in the Research Areas (RA), guided by the exemplary questions of “what new conceptual approaches do we need to achieve the goal of cognitive robot companions?” (top-down) and “how does our growing knowledge fit into the big picture?” (bottom-up). This is coupled with engineering inside each RA in order to implement and test novel ideas. In RA7, these engineering activities leave the scope of the individual lab, a deliberate decision on the part of the Cogniron Consortium as remarked during the 2005 project review [9]: “[...] the functions¹ have been chosen [...] in relation to partner interaction rather than for mainly functional or architectural reasons.”

1.2 Addressing Integration and Evaluation

The voluntary raise in scope explained in the previous section affects many of the processes required for successful engineering. In contrast to the relatively confined interaction between science and engineering inside a given lab, in RA7 they become significantly influenced by aspect of scientific agenda and technology management across the borders of research labs. The effects of this situation are notably the existence of two levels of integration and several types of evaluation.

Integration takes place

1. on a conceptual level in order to demonstrate that the results of the Research Areas in principle form a cognitive system, and
2. on a technical level where the implementation of the Key Experiments allows their use as
 - (a) instruments of science or
 - (b) prototypes of potential applications.

Evaluation is less obvious in this context, as it can take place

1. in a general research context and its established methods such as peer-reviewed publications;
2. in the context of Cogniron as an Integrated Project, which requires conceptual as well as concrete measures of success;
3. on the level of performance metrics for robot such as the ones addressed in [30]; and
4. on the level of implementation, where specifications and techniques have to match across all partners involved in a given *instance* of a Key Experiment.

¹i.e. the Cogniron Functions that serve conceptual integration of the Key Experiments

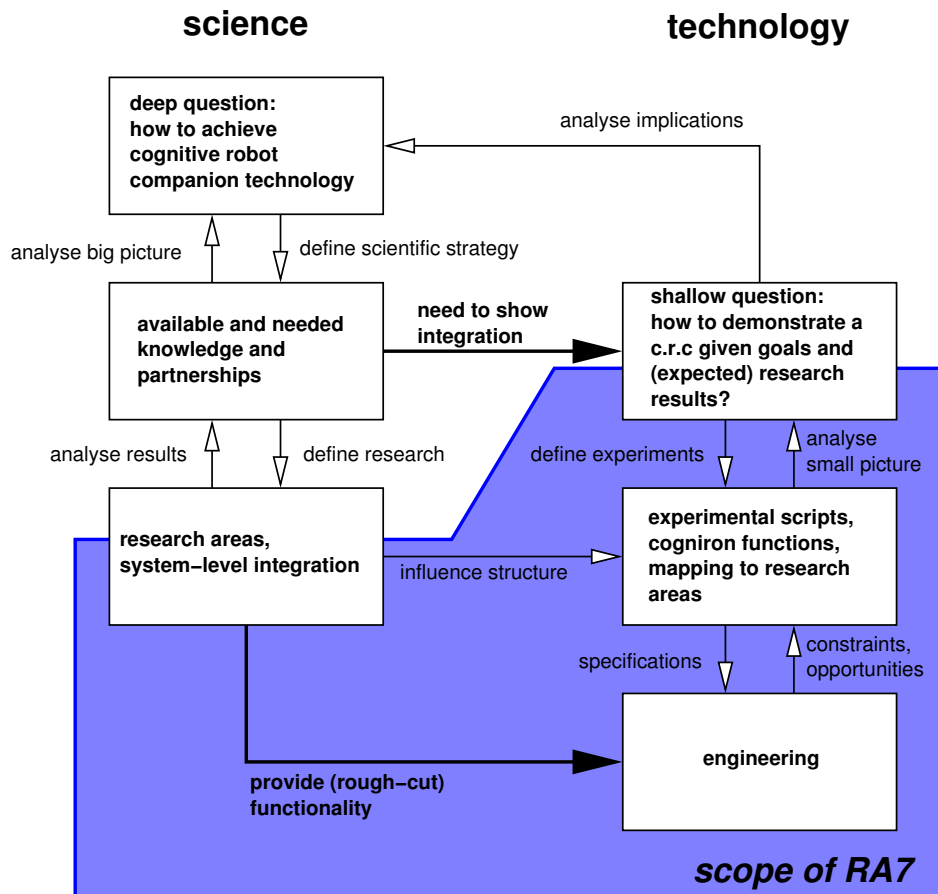


Figure 1: The scope of RA7 in Cogniron: The left side represents a view of scientific aspects, technical issues are depicted on the right. Each side has top-down as well as bottom-up mechanisms for achieving innovation. It can be seen that RA7, represented by the darker area in the lower half, straddles both sides, with two “boxes” that are on its border. The “research areas, system-level integration” box on the lower left is the source of technological transfer by providing innovative functionality to the engineering side that implements the Key Experiments, raising the issue of leadership for such transfers across the Cogniron Consortium. The “shallow question” on the upper right provides the vision that is required to guide implementation, which requires scientific leadership – again, across the Consortium. The point of this diagram is to summarize a situation that has lead RA7 to lie at the crossroads of several challenges.

The existence of these types of evaluation confronts RA7 with issues that are partly out of its scope. The peer-review evaluation of item 1 is in the domain of the other Research Activities, as long as they don't require concrete data from a Key Experiment as a whole. At the other extreme (item 4), implementation evaluation boils down to ticking off check-lists of precisely defined scenarios and technical tests that lie fully within the engineering aspects of RA7.

The end of phase 2 represents the half-time of Cogniron, and evaluation issues that concern the two intermediate levels have reached turning points. In the overall context of Cogniron (item 2), the conceptual architecture resulting from RA6 *Intentionality and Initiative* is now available for evaluation within the scope of RA7. During the second half of Cogniron there will be interesting feedback about the architecture between RA6 and RA7, which was not possible in the first half: RA7 needed an architecture to implement components relying on said architecture for their specification. Robot-level performance measurement (item 3) as a scientific question (see workshops such as [43]) is starting to provide the kind of tools that will become useful for this interaction between RA6 and RA7. Even though measuring the performance of robots is not a resolved issue, the formalism and approaches developed within that field can translate between items 2 and 4.

1.3 Progress in RA7 During Phase 2

The first three subsections present each KE in sequence by recalling its objectives and scenarios (with modifications where applicable). Then the section on software integration introduces aspects of technology transfer, concentrating on bottom-up methods. The final subsection illustrates progress with the top-down approach by introducing the specification refinements undertaken during phase 2. All these subsections are summaries with cross-references to the more detailed information contained in this deliverable.

The methodology that underlies integration progress is two-fold. The scientific vision that lead to the definition of the Key Experiments is being used in a top-down manner to incrementally refine levels of specification, from experiments to functions to interfaces and data structures. At the same time, it is neither practical to reinvent existing implementations, nor is it helpful to depend on unforeseeable research results in the specification of components, so a bottom-up tactic is added. This starts by taking stock of existing subsystems that will be required in a cognitive robot companion. Then, a framework for exchange has been provided, and is under continuous development (again relying on interfaces and data structures). This framework goes beyond technical aspects because it includes elements for fostering interchange via collaboration tools.

1.3.1 Key Experiment 1 *Robot Home Tour*

Objectives: KE1 stresses informational human-robot interaction to learn the geometry and topology of the environments and its artifacts, as well as the geometry, the identity, and the location of objects and their spatio-temporal relations.

Scenario: A human introduces to a newly purchased robot all the objects and places in a private home, insofar as relevant for later interaction.

Progress: The key experiment has integrated several Cogniron Functions (CF) and Service Functions² (SF) as described in the work plan. The individual achievements for each function can be found

²Sometimes referred to as Supporting Functions or Supporting Services.

in section 2. Progress was made for: CF-PTA *Person Tracking and Detection of Attention*, CF-DLG *Multi Modal Dialogue*, CF-ROR *Resolving Multi-Modal Object References*, CF-LOC *Dialogue or Perception Based Localization*, as well as the supporting function *Laser-Based Navigation*.

Details: Section 2 (p.12).

1.3.2 Key Experiment 2 *The Curious Robot*

Objectives: KE2 stresses the perceptual skills of the robot to acquire knowledge about the locations and properties of objects and their spatio-temporal relations.

Scenario: KE2 employs two scenarios. In the first instance, a robot notices that it can assist a human in a given situation, whereupon the robot decides to act without being explicitly asked for its help. The second KE2 scenario presents a robot exploring new objects in its environment, subsequently interacting with a human to name the objects to complete its newly acquired knowledge.

Progress: In order to allow simultaneous work and proceed toward partial integration without waiting for the target mobile manipulator to be fully functional, it was chosen to build separate sub-experiments on three different platforms:

- Experiment KE2-1: Object modeling and handling by a mobile manipulator
- Experiment KE2-2: Toward human-friendly navigation
- Experiment KE2-3: Curiosity and learning

The efforts have been essentially devoted to the development and integration of the basic ingredients on which KE2 experiment will be constructed. A methodology for integration has been defined and several tools have developed and/or deployed to allow the incremental development and integration of the different robot functions. Partial integration has also been performed. The first Cogniron services have been encoded and implemented on the platforms. Substantial ingredients are now available that will allow to build and demonstrate several abilities involved in KE2.

Details: Section 3 (p.17).

1.3.3 Key Experiment 3 *Learning Skills and Tasks*

Objectives: KE3 stresses the learning and reasoning capabilities for the robot to acquire knowledge about goals and tasks employing the example of laying out the table.

Scenario: KE3 also consists of two scenarios. In the first, the robot learns to discriminate between relevant and irrelevant behavior for manipulating objects, by observing repeated demonstrations which differ in the order of manipulations as well as object positions and displacements. The second KE3 scenario stresses life-long and context-based aspects through incrementally learning to serve a guest by repeated annotated demonstrations. The latter thus concerns *tasks*, whereas the former concentrates on *skills*.

Progress: The functionalities of this key experiment that were to be implemented are defined in the work plan of RA7 for the second phase of the project. The following Cogniron functions were implemented in this phase: CF-LIF *Learning Important Features* and CF-RG *Learning to Reproduce Gestures* at EPFL in cooperation with UH (see deliverable RA4); as well as CF-LCT *Learning Complex Task Descriptions*, CF-GR *Gesture Recognition*, and CF-OR *Object Recognition and Modeling* in collaboration between UniKarl and IPA. The functionalities were implemented on three platforms: a humanoid platform at EPFL for imitation learning, and for task learning on a mobile robot with manipulator at IPA and a sensor setting for motion tracking at UniKarl.

Details: Section 4 (p.23).

1.3.4 Software Integration

In section 5 we present how the notion of Cogniron Functions (CF) has led to the definition of Cogniron Service Functions (SF) and data structures. These services led to structuring technology transfers using a module market on a common exchange platform on the Cogniron Wiki. A Wiki is a web-based collaboration tool where documents can easily be edited and shared by a group of people, with a great potential for reducing administrative overhead if a certain amount of discipline can be expected from the contributors.

The notion of services provides a technical framework for enabling bottom-up software integration to meet the top-down process of conceptual integration and CF specification. Services are grouped by module on the Wiki, a snapshot of this ongoing work is given in appendix D.

1.3.5 Refined Specifications

Refined Environment and Script Specifications To have a clear definition of the scenarios in which the key experiments are conducted, conditions for all key experiments were collected and compiled. The following features, describing the environments of the experiments, were defined: Physical conditions (light and sound); rooms (e.g. living room, hall); locations (e.g. desk area); furniture (e.g. table, chairs); objects (instances, properties, and positions); and humans (activities and dialogue).

A common template was used for each key experiment to ease extracting of common properties. Detailed scripts were written for the key experiments, describing the tasks, functionality and overall control loops. This clarifies many of the open issues that subsisted after the first set of deliverables [2, 3, 4, 5]. The resulting updates of the RA7 workplan for phase 2 is summarized in appendix E (KE1: table 7 (p.85), KE2: table 8 (p.89), KE3: table 10 (p.91)).

Refined Software Component Specifications Starting from the specification document [2], a project wide questionnaire was designed to further detail the Cogniron functions in terms of: Input and output (data formats, sources and destination such as sensors or actuators or other components); internal modes; required computational resources (e.g. processing power, storage capacity); type of function (interaction, perception, learning, motion generation, or autonomous decision making); as well as testing methods.

The questionnaire results are summarized in appendix C. Note that ongoing work in making the bottom-up software integration meet the top-down specifications can potentially influence some of the details presented in the above. The current state of service definitions, grouped by module, are presented in appendix D for comparison.

1.4 Conclusion and Outlook

Integration As shown in the detailed sections of this deliverable, integration is progressing mostly according to plan. Temporary deviations due to considerations of concurrent engineering (several robots for a given KE) and incremental development (use of mock-up components) are considered normal courses of action in this context. They are not expected to threaten the completion of RA7 objectives. Recall that the Key Experiments have been designed to allow evaluation of partial instances of the conceptual architecture which results from RA6 (see its deliverable [8]). Thus, conceptual integration and evaluation of the whole Cogniron architecture lie outside the scope of RA7 as presented in sections 1.1 and 1.2.

Testing on a Functional Level Defining performance rates for all Cogniron Functions is well under way, the state of advancement is documented in appendix C. This allows testing on the engineering level “below” the one of the Key Experiments. As far as reasonable, such testing is aimed at using existing test data bases such as COIL [42] for object recognition. We will continue to keep track of and refine these test criteria in a coherent manner as part of RA7 activities in the next phase. However, we are aware that this is not sufficient in all cases because Cogniron Functions tend to emphasize open-endedness.

Outlook on Evaluation and Testing On the Key Experiment level, the detailed scripts allow evaluation to proceed by “checking off” robot performance in terms of specific behaviours. The scripts were defined in an incremental manner and the Cogniron Functions were derived from them. Along with the environmental specifications this is quite more rigorous than methods commonly used for demonstrating autonomous robots. At the same time it is clear that this approach can neither evaluate the open-endedness mentioned above, nor the requirements that pertain to upcoming results of RA3 *Social Behavior and Embodied Interaction*. It appears likely, from an engineering perspective, that some concept such as “human-like” can not be expressed in numbers, and RA3 rather focuses on the human perception of the robot than evaluating against some unknown intelligence measure. Thus, in conjunction with RA3 and the architectural feedback from RA6 mentioned in section 1.2, the perspective of RA7’s evaluation activities will undergo a shift from engineering to science during the second half of Cogniron.

2 Key Experiment 1 *Robot Home Tour*

2.1 Role of KE1 in Cogniron

The role of KE1 was precisely defined in the *Periodic Activity Report* of 2004 [10]. Recall that the main objectives are to demonstrate:

- The outcomes of RA1 demonstrating a multi-modal dialogue;
- partial results of RA2 related to human tracking, deictic gestures and object references;
- partial results of RA3 with respect to the socially acceptable interaction;
- partial results of RA5 targeting the learning and representation of locations;
- as well as partial results of RA6 relating to intention attribution.

And the main scientific issues tackled by KE1 are:

- Dialogue-oriented cross-modal information processing and representation;
- interactive learning of objects and locations;
- adaptation of robot behavior to different types of users.

The CFs that are relevant to KE1 are integrated as modules inside a flexible infrastructure presented in [15]. Figure 2 gives an overview based on the implementation architecture. The detailed setup and script can be found in appendix E, in particular table 6 (p.84) shows the relationship between KE1 and CFs, RAs, and Cogniron partners. The link to developments in RA6 is additionally depicted in figure 3, which shows how KE1 components match the cognitive architecture presented in the deliverable of RA6.

2.2 Status of KE1 at T0+24 (December 2005)

During the last 24 months, it was possible to realize the Cogniron and supporting functions as follows, integrating them into the robot *Biron* for improved interaction capabilities [16]. Recall that KE1 is implemented on *Biron* (depicted in figure 4) at UniBi.

2.2.1 CF-PTA *Person Tracking and Detection of Attention*

The existing CF-PTA was extended to a more robust detection of potential communication partners and to enhanced interaction capabilities. This was done by extending the frontal face detection to cope with different gazing directions determining whether a person is directly looking at the robot or to another direction. If a person is fixating the robot the robot will greet the person to get his attention. Furthermore the CF-PTA was extended to enable interaction with multiple persons, even when no wide range sensor like the laser range finder is available (see RA2). A first version of a person memory stores not only the last known position when a person gets out of the robot's view, but also person specific data after the communication partner has been identified. This also enables storing individual preferences like different social spaces. Based on the results from UH standard distances for following and interaction have been implemented.

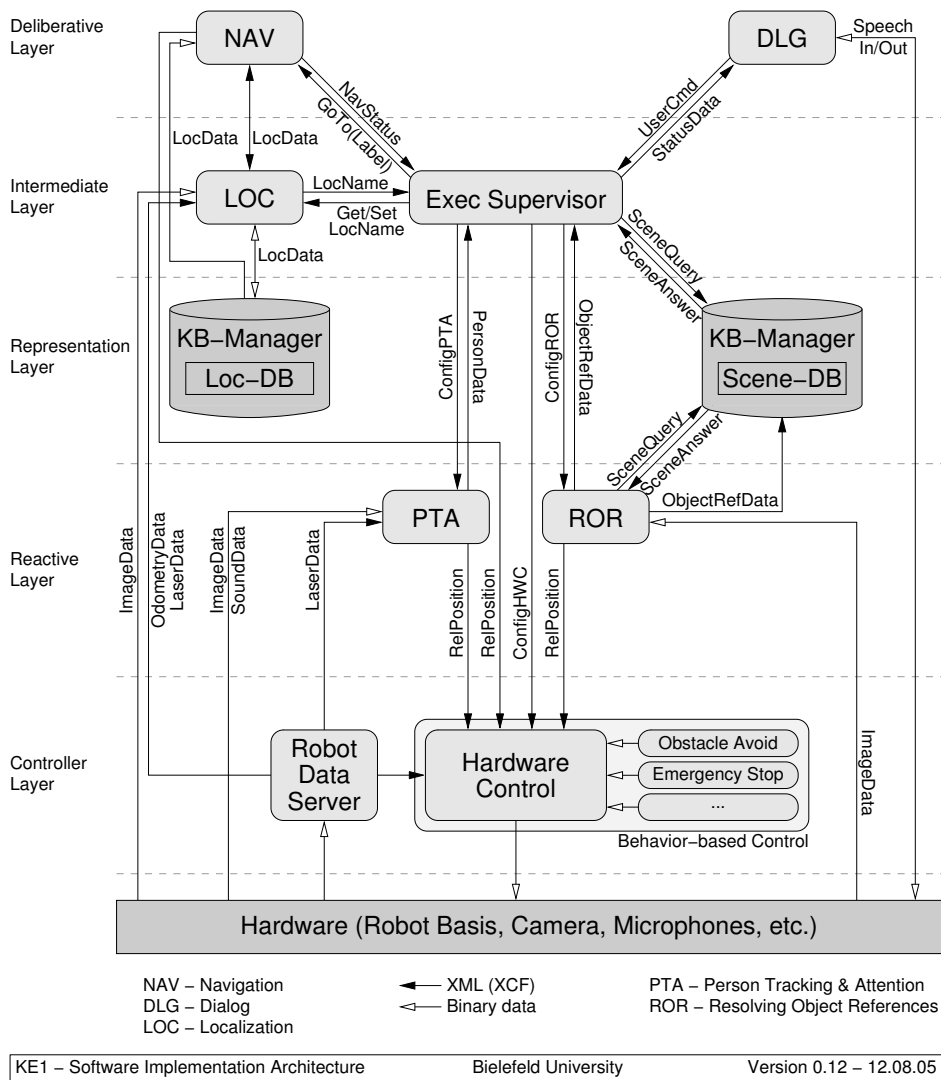


Figure 2: The software implementation architecture used for Key Experiment 1.

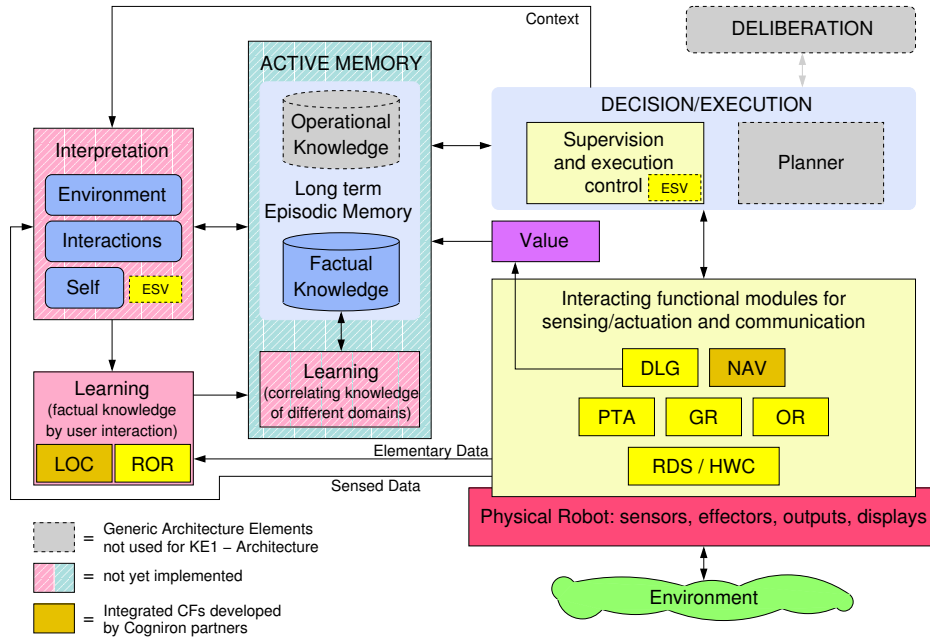


Figure 3: The components of KE1 match the cognitive architecture developed in RA6

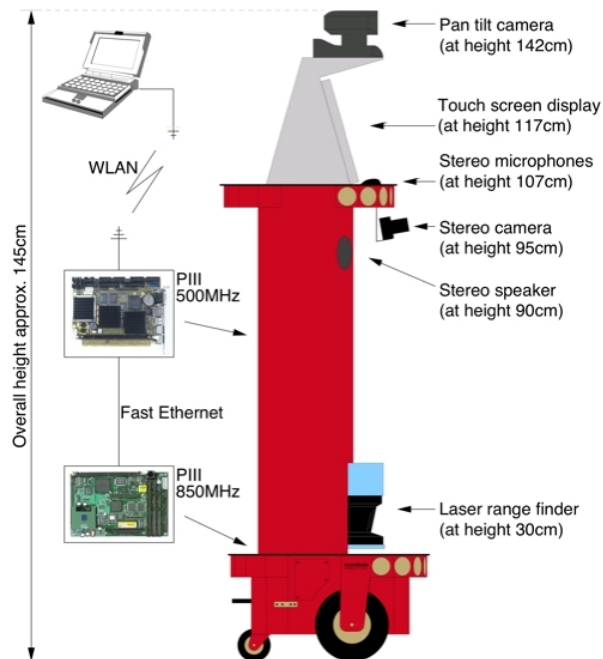


Figure 4: Key Experiment 1 is being implemented on the mobile robot Biron at Bielefeld University.

2.2.2 CF-DLG *Multi Modal Dialogue*

The reimplementation of the dialogue as described more detailed in RA1 can operate with a higher language complexity supporting verbally specified cues for object references or location handling. The dialogue is able to help the user in interacting with the robot. For example, the robot tells a person that it can not look at an object and follow a person at the same time. The robot can also explain how it can be made to first follow and then look at an object. As user studies show a verbally more active robot is preferred to a less talkative one, so additionally the robot excuses itself for repeatedly misunderstood phrases or tells the user that it likes the interaction.

2.2.3 CF-ROR *Resolving Multi Modal Object References*

The CF-ROR is using the SF "2D Gesture recognition", "Simple object recognition" extended by color based object learning and "Scene Model". The robot now detects deictic gestures with its second camera mounted on a height of 95cm, and will align its main pan tilt camera to the position a person points at. Given a name and an attribute, e.g., a color by the CF-DLG the robot will combine this information with the object image from the pan tilt camera image using the "Simple object recognition". Due to limitations of processing power this is at the moment running on a desktop computer directly connected to the video cameras of the robot. A video of this demonstration is available³.

2.2.4 CF-LOC *Dialogue or Perception Based Localization*

After developing general interfaces in cooperation with UvA and LAAS a Skeleton module delivered by UvA is integrated, simulating CF-LOC and its SF Navigation. Porting of the real LOC and Navigation will follow soon using the advantages taken out of the extensive phase of testing and interface discussion. Because of the general interface structure a simulated LOC using a real Navigation (as described in the following section) was integrated and demonstrated on the robot. A video of this demonstration is available⁴.

2.2.5 SF *Laser-Based Navigation*

This section presents a path planning and obstacle avoidance approach for mobile robots ported to *Biron*. The approach was originally developed at EPFL for the Robotics@EXPO.02 mass exhibition [35]. This work has been published as part of a thesis [33]. The system has proven its value during a 5 month operation of eleven *Robox* tour guide robots in a real-world application, a very crowded exhibition. Three techniques (DWA [31], elastic band [34], NF1[32]) have been integrated into a system for tour-guiding that performs smooth motion efficiently.

The *DWA* generates actuator commands such that the robot does not collide with obstacles, the commands do not violate the dynamic capabilities of the actuators, and the robot follows the planned path. The *elastic band* is responsible for path representation and adapting the plan to the robot's movement and changes in the environment. Creating an initial plan from the robot's current position to the goal is the task of the *NF1*, a simple grid-based planner.

The main sensor input for the motion planner is laser scanner data. The robot stops immediately if a laser reading indicates that an object is inside the robot's outline, otherwise the sensed obstacles

³The video is on the CD submitted with the deliverables 2005 (section RA7) and can also be downloaded from <http://www.cogniron.org/wiki/CognironDeliverables2005> (attachment D7.2.2_ROR_DivX_mp3.avi).

⁴Also on the CD, downloadable from <http://www.laas.fr/~rolo/media.html>

influence the evasive actions. A set of convex polygons models the robot shape. All calculations are based on the latest sensor data. Higher level modules on Biron will provide topologically correct sub-goals (see CF-NAV / CF-LOC).

2.3 Conclusion and Outlook

As described in more detail in the workplan of KE1 for the next phase the focus will be on the exchange of preliminary or simulated CF and SF by more intelligent CF developed within the project, making the overall system more robust and adding new interaction capabilities using memories.

3 Key Experiment 2 *The Curious Robot*

3.1 Role of KE2 in Cogniron

The objective of KE2 relates to robot curiosity triggered by a drive for learning more about the environment. It also includes issues related to the capacity for a robot to take initiatives in the presence and in interaction with humans. The underlying scientific issues are the following:

- Object learning and recognition (RA5);
- Interpretation of human activities and posture (RA2);
- Close interaction with humans (RA3);
- Intention attribution (robot intention should be legible for humans) (RA6);
- Situation understanding, decision making, collaborative problem solving (RA6);
- Cognitive architectures (RA6).

Two separate scripts are developed, one emphasizing robot curiosity, and the other robot initiative and intentionality. The first script describes how a robot builds a model of a newly discovered object through sensing and manipulation. The second script describes how a robot interprets a situation that requires its intervention for fetching an object and handing it to a person.

During phase 2, the main focus in KE2 was on setting up the necessary material for demonstrating the scenarios, and integrating parts of the software developed in the various RAs mentioned above. In order to ease simultaneous work and proceed toward partial integration without waiting for the target mobile manipulator to be fully functional, a concurrent integration approach was chosen to build partial experiments on three different platforms:

- concerning **Robot Initiative and Intentionality**:
 - KE2-1: Object Modeling and Handling by a Mobile Manipulator
 - KE2-2: Toward Human-Friendly Navigation
- concerning **Robot Curiosity**:
 - KE2-3: Curiosity and Learning

Each experiment corresponds a partial integration of the targeted functions involved in the global KE2 scenario. The three platforms share the underlying software architecture and tools⁵.

No integrated demonstration was planned for this year. However, a number of the target CFs have been partially implemented. We report here below on their current state and as well as on a set of associated supporting services.

Figure 5 illustrates the role of the different CFs with respect to the generic architecture elaborated in Cogniron. Partial instances of this architecture will be realized in order to build and run partial experiments (cf. Annex A).

⁵<http://softs.laas.fr/openrobots/>

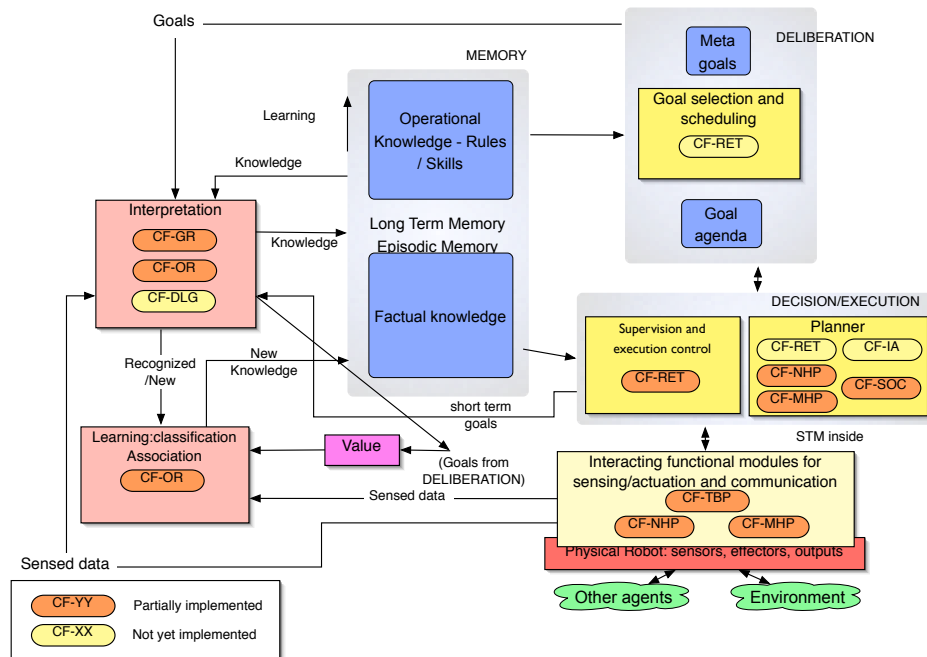


Figure 5: Software architecture for the demonstrator platforms of KE2

3.2 Status of KE2 at T0+24 (December 2005)

3.2.1 CF-MHP Manipulation in Human Presence

Jido, the mobile platform chosen for implementing KE2-2, is now operational with its supporting services (perception, arm and hand control, locomotion, localization and navigation). Modeling and localization of objects is not yet entirely operational. The grasp planner is capable of generating trackable trajectories and we are working on exteroceptive sensor feedback. A video⁶ showing a trajectory computed by the grasp planner and executed by Jido is available.

A soft motion trajectory planner was implemented to control the arm with a flexible control law for visual and force control. The grasp planner uses geometric properties (center of mass and inertial directions) to guide a random generator of grasp positions that are filtered and evaluated to choose one that is free of collision and mechanically acceptable. This planner was extended for non-convex objects and to plan collaborative tasks like handing over an object. Videos⁷ showing illustrative simulation results are available. This is a first step towards implementing human-robot interactive manipulation scenarios for RA3.

3.2.2 CF-OR Object Recognition

The object recognition function has been implemented at LAAS within KE2.3. In order to perform object learning and recognition, a multi-layered neural network is trained on-line in an unsupervised way from stereo camera data. The resulting representation of the object is view-based: a collection of neural detectors learns to recognize the different views of the object. In addition, temporal coherence

⁶Video is on the CD submitted with the deliverables 2005 (section RA7).

⁷Videos are on the CD submitted with the deliverables 2005 (section RA7).

between consecutive views is used to autonomously create and pool together sets of detectors. Finally, a high-level neural detector is able to recognize a particular object from various points of view.

3.2.3 CF-NHP *Navigation in Human Presence* and CF-SOC *Socially Acceptable Interaction with Regard to Space*

User studies done at UH in the context of CF-SOC have driven the specification of CF-NHP. These studies show, for instance, that humans have preferences about approach directions of a robot. This is one of the properties that were taken into account during the design of the navigation planner. The resulting trajectories reflect these preferences of the human user. Videos⁸ showing simulation results are available.

CF-NHP is implemented at LAAS on the robot Rackham. The navigation planner is now capable of leading the robot on human-friendly trajectories. Two supporting services, a human localization service and a motion execution service, are used to handle the input and output data of the planner. Although the implementation results are not mature enough to report, simulations and processing time measurements of the navigation planner are encouraging. The detailed implementation process and the inner working of the planner are explained in appendix B and Deliverable 3 [6].

3.2.4 CF-TBP *Tracking of Human Body Parts for Observation*

Functions dedicated to human detection and tracking using visual primitives have been implemented at LAAS for CF-TBP. For short, medium and long range 2D tracking of humans and body parts, a particle filtering framework has been developed. It has enabled the principled design and implementation of 2D visual trackers relying on color and/or shape and/or motion cues as well as on diverse particle filtering schemes. The particle likelihoods defined from the visual cues fusion/combination, together with the importance functions specified from visual detectors, were thoroughly evaluated on images grabbed in a robotics context. Criteria such as discriminative power, precision, and time consumption have been studied. Concerning the estimation engine, the standard particle filtering formalism as well as alternative schemes were evaluated in order to check which ones best fulfill the requirements of the three following human-robot interaction modalities: head tracking, human tracking, and monitoring. For each of these, a separate visual tracking strategy was selected. All the modalities have been implemented in a module called ICU. Ongoing work addresses appearance-based 3D human posture estimation. Two kinds of videos⁹ are available, one showing an example of integration and the others demonstrating various particle filtering strategies.

3.2.5 CF-GR *Gesture Recognition*

Some results on 2D communicative gestures interpretation have been achieved at LAAS. The approach has been devoted to the classification of hand configurations and fronto-parallel motions in the video stream. *Mixed-State Condensation* has been proposed to detect the most likely hand posture and canonical motion model. This ensures an automatic switch between multiple template motions in the tracking loop. The aim is to get a compact representation of a gesture through a sequence of hand configurations and motions. The symbolic gestures tracking and recognition system was enhanced through the definition of a new likelihood function, fusing shape and color cues, which significantly raises the recognition rate of static configurations. An enhanced *Jump Markov Particle Filter* is being

⁸Videos are on the CD submitted with the deliverables 2005 (section RA7).

⁹Videos are on the CD submitted with the deliverables 2005 (section RA7).

implemented, which is expected to lead to better performances for dynamic gestures segmentation. The hand configuration recognition system will be implemented on Rackham in KE2 to allow gesture-based interaction.

3.2.6 CF-RET Reasoning About Tasks and its Own Abilities

A supervision framework has been developed at LAAS for CF-RET. We have implemented an agent-based supervision system based on joint intention theory. It deals with tasks in terms of (i) individual tasks (only the robot is involved), (ii) joint tasks (the robot and another agent are involved), and (iii) activities that corresponds to atomic low level functionalities. Each task is defined by a plan and associated monitors. A plan corresponds to a succession of sub-tasks and/or activities. Monitors detect whether a task is unachieved, achieved, impossible, irrelevant, or stopped. Consequently, the system can be controlled at different levels at the same time. The system is able to react by applying level-specific solutions and, when necessary, propagating events upwards or downwards. Videos¹⁰ are available.

In the current implementation, the supervisor is written in Open-PRS¹¹. Task plans are hand-coded in a pre-defined task library and only the robot is able to propose a task. However, the supervisor is designed in view of future extensions involving on-line task planning. Details are given in appendix B and in Deliverable 6 [8].

3.2.7 Supporting Services

The following supporting services have been implemented at LAAS:

- Jido Arm Control with:
 - `qarm`: interface with Mitsubishi PA10-6C arm
 - `xarm`: Cartesian control based on an on-line soft motion trajectory planner
 - `collide`: collision checker
- Jido Hand Control with:
 - `fingers`: interface fingers sensors (Force Sensor Resistors and strain gages for contact forces and finger position)
 - `force`: interfaces a six-axis force sensor
 - `graspCtrl`: gripper control (not yet entirely implemented)
- Base Localization with:
 - `sick`: interface with the SICK laser sensor
 - `segloc`: laser-based localization
 - `pom`: probabilistic pose fusion (from `segloc`, the odometry and the gyro)
- Jido Motion and Navigation with:
 - `JLOCO`: basic motor control and odometry

¹⁰Videos are on the CD submitted with the deliverables 2005 (section RA7).

¹¹<http://softs.laas.fr/openrobots/tools/openprs.php>

- SFL: “Sunflower” mobile robot library from EPFL. It provides path planning and obstacle avoidance.
- Rackham Obstacle Avoidance and Navigation with:
 - aspect: laser data integration to overcome the laser limitations (180 degrees)
 - NDD: Nearness Diagram Navigation ([40])
- Human Positioning with humPos: produce an input on a person’s position in the environment using laser information

3.2.8 Status of KE2 Script *Robot Initiative and Intentionality*

The definitive platform on which KE2 will be implemented is Jido (see figure 13). However, considering that this platform has been purchased in February 2005 and required a certain amount of hardware and software development before effective use, we decided to continue working on Rackham (see figure 20) in parallel with Jido development. Note that all software is development within the same underlying architecture and toolset, which is largely platform independent.

KE2-1: Object Modeling and Handling by a Mobile Manipulator (Jido) KE2-1 has been designed to implement generic object modeling and manipulation abilities that are planned to be used in KE2.

Work done in 2005 :

- Equipment of a new mobile manipulator
- Installation of software integration tools
- Definition of a set of Cogniron services (and their interaction) related to object modeling and manipulation
- Adaptation of a set of supporting services (localization and motion)
- Development and Integration of new supporting software
- Development of a grasp planner
- Development of a new control law for manipulation
- Development of a stereo-based object modeling service

All essential ingredients are now available to build and demonstrate an autonomous and generic ability to model and handle objects.

KE2-2: Toward Human-Friendly Navigation (Rackham) KE2-2 has been designed to investigate and illustrate mobile robot navigation in the presence of humans and to approach persons in order to interact with them or to hand over objects. The robot must be able to detect humans in its vicinity and to interpret their trajectories, in order to avoid or interact with them. One contribution focuses on the development of people detection and tracking modalities from the video stream of the on-board color camera, each modality being suited to a specific H/R interaction distance. The other contribution focuses on the illustration of a Human Aware Motion Planner.

Work done in 2005:

- Definition of a set of Cogniron services (and their interaction) related to motion in the vicinity of humans
- Development and integration of a “Human Aware Motion Planner” (CF-NHP)
- Development and integration of a first version of robot supervisor specially dedicated to human-robot interactive task achievement (CF-RET)
- Development and integration of a set of vision-based human detection and tracking services (CF-TBP)
- Development and integration of several supporting services

The overall experiment has not yet been demonstrated. However, all essential ingredients are now implemented.

3.2.9 Status of KE2 Script *Robot Curiosity*

Experiment 3: Curiosity and Learning

Work done in 2005:

- Development of the neuron activity monitoring and network creation software: Assemblee.
- Validation of the object learning and recognition algorithms (IJCAI-05).
- Development of dedicated neural networks for positions and actions representations.
- Preliminary work on sensory-motor association.

Results on learning composed actions are encouraging and we now investigate the learning of sequenced actions. During phase 3, we will focus on the action selection and production level.

3.3 Conclusion and Outlook

Phase 2 has been essentially devoted to the development and integration of the basic ingredients on which KE2 will be constructed. A methodology for integration has been defined and several tools have been developed and/or deployed to allow the incremental development and integration of the various robot functions. Partial integration has also been performed. The first Cogniron services have been encoded and implemented on the platforms. Substantial ingredients are now available that will allow to build and demonstrate several abilities involved in KE2.

4 Key Experiment 3 *Learning Skills and Tasks*

The Key Experiment “Learning Skills and Tasks” stresses the learning and reasoning capabilities for the robot to acquire knowledge about goals and tasks employing the example of laying out the table and serving a guest. At the current stage it is planned to demonstrate and assess the following skills which are implemented on a robot platform:

- Learning goals from observations (RA4, RA5, RA6)
- Reproduction of the goal for arbitrary starting conditions (RA4, RA6)

Learning activities from human demonstrations can be done at several, at least two, different levels of granularity of the learned and reproduced entities, each with different complexities. For this reason, it was decided to duplicate and demonstrate the learning cognitive function within two scripts, implemented on two different platforms. The first script is concerned with Learning Skills: Arranging and interacting with objects, while the second script is concerned with Learning Tasks: Serving a guest. The scripts and their relation to the Cogniron function are described in full detail in deliverable D.7.1.1 [2].

The remainder of this part is organized as follows: Section 4.1 gives an overview on the relationships of Key Experiment 3 with other Cogniron sub-projects. The current status of the implementation of each Cogniron Function involved and the two scripts of Key Experiment 3 is summarized in section 4.2. Finally, an outlook on the work to be done is given in section 4.3.

4.1 Role of KE3 in Cogniron

The Key Experiment 3 plays an integrative role within the whole project, that pools the results gathered in research activities RA2, RA4 and RA6. Its general aim is to stress the issue of the integration of learning algorithms, methods and components resulting from RA4 with several other components, e.g. perception (RA2) and object recognition (RA5). The methods and algorithms developed in the RAs are implemented in several Cogniron Functions that provide software modules that are demonstrated and evaluated in the context of the two KE3 scripts “Learning Skills: Arranging and interacting with objects” and “Learning Tasks: Serving a guest”.

The Key Experiment 3 activity deals essentially with the design and hardware implementation of the KE3. The core of the scientific approach underlying the development of each of the Cogniron Functions, necessary for the implementation of the present KE are reported in the deliverables of the related RAs. The connection between the demonstrated Cogniron Functions and the other components of the Cogniron Architecture are shown in figure 6. Note that CF-LCT and CF-OR appear two times in the diagram, as they are apparent in the scripts in different modes: a learning mode and a reasoning/recognition mode respectively. This is implemented in two different modules of each of those Cogniron function.

4.2 Status of KE3 at T0+24 (December 2005)

This section describes the current status of the work conducted within KE3. It is organized along the Cogniron Functions involved in the two KE3 scripts, namely CF-LIF, CF-RG, CF-LCT, CF-TBP, CF-ACT, CF-GR and CF-OR. The overall integration of these Cogniron Functions into the two systems used within KE3, the HOAP platform at EPFL and the IPAbot at IPA, is described in the two additional subsections 4.2.8 and 4.2.9.

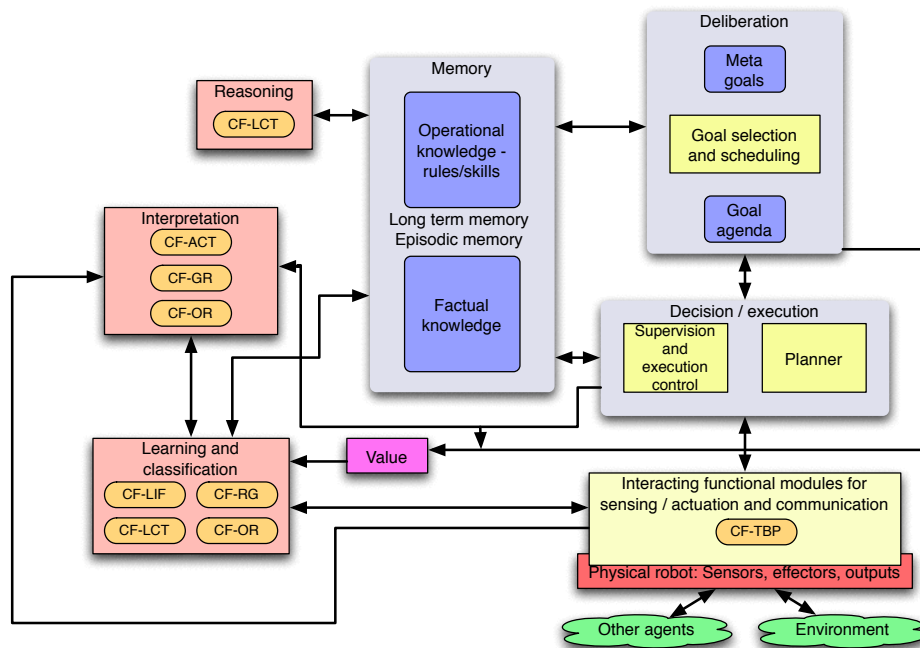


Figure 6: Software architecture for the demonstrator platforms of KE3.

4.2.1 CF-LIF Learning Important Features of a Task

For the robotic companion to be able to effectively learn from the observation of humans performing a given task, it must be capable of extracting what is relevant and discarding what is irrelevant in a given demonstration. This is the purpose of CF-LIF.

The implementation of CF-LIF is driven by the scientific result from the work conducted at EPFL and UH within WP4.1 and WP4.2. It is reported in Part 1 and 2 resp. of the RA4 joint deliverable D4.4 [7].

The work conducted at UH within WP4.2 towards a characterization of the space of effect, state and action metrics drives the development of algorithms for automatic extraction of features of a task [23, 24].

The work performed at EPFL contributes to CF-LIF by developing an architecture for automatic extraction of the important features of a task [21, 22]. The features that are statistically invariant across the various demonstrations are deemed as the relevant features that constitute the task. This is driven by the assumption that features occurring more often in demonstrations of the same task are more likely to be of higher relevance to the successful accomplishment of the task.

In phase 2, a probabilistic method was developed, based on the encoding of gestures using Principal/Independent Component Analysis (PCA/ICA) and Hidden Markov Models (HMM). It was used for extracting the relative importance of reproducing either the gesture or the specific hand path in a given task. This, then, determines a metric of imitation performance that can be used by CF-RG to determine the quality of an attempt to reproduce a task on the robot companion.

4.2.2 CF-RG Learning to Reproduce Gestures

The robotic companion in Cogniron should be able to replicate a demonstrated task by its user (imitate), although its shape, embodiment, abilities and affordances will differ from those of a human (i.e.

the robot needs to address the correspondence problem).

CF-GR builds on results obtained within WP4.2 at UH and EPFL which can be found in part 2 of the RA4 joint deliverable [7].

One contribution to CF-RG is the development of the JABBERWOCKY system done at UH [23]. JABBERWOCKY is a system that uses captured motion data of a human demonstrating a task to produce action commands that if executed by an imitating agent result in achieving corresponding effects. JABBERWOCKY is able to generalize across dissimilar initial object configurations given the task subgoal granularity and appropriate effect metrics, generating action commands targeted for multiple imitator platforms, both simulated in software and in hardware. Additionally, it presents a novel generic approach to the correspondence problem, using body-mapping for cases of state and/or action matching formalized via correspondence induced metrics [24].

Also, the work conducted at EPFL provides methods for optimizing locally the path to be reproduced according to the task's metric (learned in CF-LIF), taking into account the features of the task, as well as the body of the robot and external constraints.

A video that illustrates the "what-to-imitate" and "how-to-imitate" issues in a task consisting of moving a chess piece on a chessboard is available¹². The constraints are extracted statistically by demonstrating the same task five times with different initial configurations. The task is then reproduced by the robot with configurations that have not been observed during the demonstrations.

4.2.3 CF-LCT *Learning Complex Tasks*

Learning and Reasoning are two of the most important capabilities of a Cognitive Robot Companion. CF-LCT tackles both of these abilities in a unifying view, stressing the close interconnection between both of them. It equips the robot with the capability to (1) acquire new task knowledge by the observation of a single task demonstration performed by a human and the recognition of vocal or deictic comments given by the user and to (2) complete this task knowledge in an incremental manner when more demonstrations or comments become available.

The implementation of CF-LCT is driven by the scientific results of work performed by UniKarl in WP4.3. This work is reported in part 3 of the RA4 joint deliverable D.4.4 [7].

The first part of CF-LCT initially manages the acquisition of raw data from dedicated sensors, namely a magnetic field position and orientation sensor and a data-glove for each of the user's hands. In several processing steps this data is denoised, segmented into elementary operations and restructured into the hierarchical task representation found in WP4.3. This Module has been implemented and tested in the second phase on the UniKarl hardware and the IPAbot platform. It offers an interface that provides the task knowledge gathered during the learning process to a robot companion system in XML format.

After the initial task acquisition step learning is not done. During the whole life time of a robot companion additional reasoning is applied to gain insights from the holistic consideration of the complete task knowledge a robot. The goal is to find similarities of new task demonstrations with already demonstrated and acquired tasks [25]. Once these are found, several reasoning methods will be applied in order to exploit the different similarities and gain new insights. After that the resulting new or adapted task knowledge will be incorporated back into the task knowledge database to be available for execution or further improvement.

A first reasoning module was implemented and tested during the second phase. This contained the reconstruction of task precedence graphs from several demonstrations of the same task offering the

¹²The video is on the CD submitted with the deliverables 2005 (section RA7)

possibility to reorder actions at execution time [26]. This relies on the work done in WP4.3 during phase 2 and is fully implemented and extensively tested as yet.

4.2.4 CF-TBP *Tracking of Body Parts*

In order to learn skills and tasks from the observation of human users the robot needs to perceive the human and especially the actions he performs. This functionality is provided by CF-TBP.

The Cogniron Function Tracking of Body Parts (CF-TBP) is under development in RA2. It focuses on tracking of one human in close distance to the robot, determining the body pose of torso and extremities (including legs) in 3D over time [27]. This information is required and useful for different purposes, like activity classification, haptic interaction or observation of human actions. The latter is an important feature of KE3: For both scripts, comprehensive observation capabilities are required.

Within the second phase, several additions and improvements have been made for CF-TBP. The developed fusion methodology and framework for tracking are implemented and run in real-time; an improved joint model has been integrated for the 3D geometric articulated human body model; and different tracking modalities have been successfully combined to gain higher robustness in tracking. All required interfaces for integration of CF-TBP have been defined, and integration of the first version of the TBP tracking system on the KE3 robot is currently in progress.

4.2.5 CF-ACT *Recognition of Human Activities*

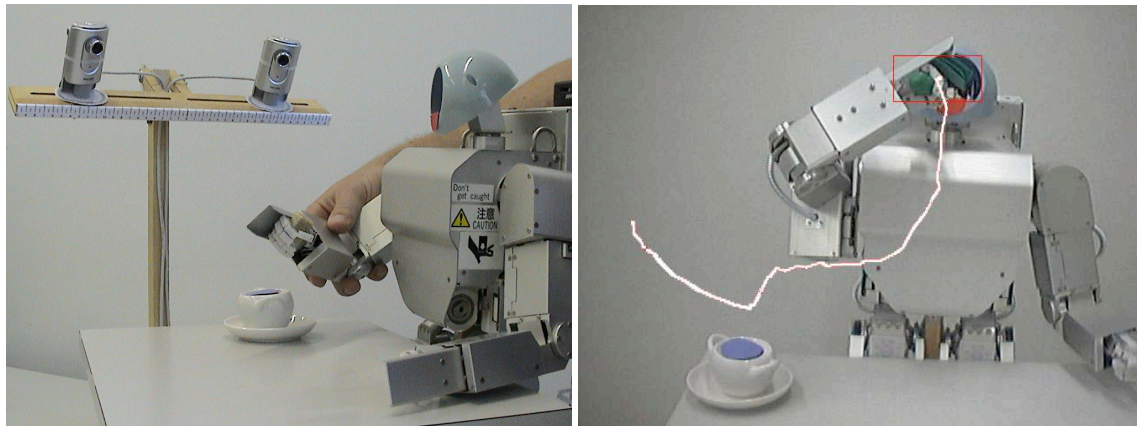
Learning from human users involves to a large degree the recognition of the current intention of the user. This in turn relies on the successful determination of the particular sequence of actions the human performs. This data is obtained from the CF-ACT.

CF-ACT (recognition of human activities) is also developed by RA2 and aims at classification and recognition of human activities of persons who are interacting with the robot [28]. Which activities need to be recognized is currently a topic of research between RA2 and RA3. At the moment, a test set of activities for system evaluation has been trained and can successfully be recognized. These activities include walk, wave, handshake, bow and other activities humans perform during interaction. CF-ACT has so far been partially integrated in the tracking and fusion framework, which in turn is currently being integrated in the KE3 robot. At the moment, the activity recognition can be used at 1/3 Hz without significantly affecting the tracking functionality, but efforts for gaining higher real-time capabilities still need to be carried out.

4.2.6 CF-GR *Gesture Recognition*

For commanding a robot companion gestures are an important input modality. Therefore the classification of gestures is an important preliminary for the selection of objects and places.

The CF-GR (gesture recognition) has large overlap with CF-ACT. The current set of activities which can be recognized within CF-ACT also contains e.g. "pointing" and "waving", which are gestures both targeted at an interaction partner. Nevertheless, special efforts have been made to tackle the important area of gesture recognition: For communicative gesture interpretation, work has been devoted at classifying hand configurations depending on the number of open fingers and of fronto-parallel motions from 3D image sequences. Studies have been carried out together with all KE leaders to identify important gestures and discriminating features. The results have been (and will be) used to improve the gesture recognition algorithms.



(a) Teaching of the robot through kinesthetics (moving the robot's arm through the task)

(b) Experimental results of reproduction of the demonstration

Figure 7: Setting and demonstrator platform of Key Experiment 3, script 1.

4.2.7 CF-OR Object Recognition

The Cogniron Function object recognition has been implemented at IPA with the aid of people from UniKarl. UniKarl contributed the superquadric models used for 3D object modeling. At IPA a first detection and pose estimation algorithm based on superquadric models was implemented on the IPA platform [29]. Also an appearance-based algorithm that was developed as part of RA5 (object modeling) was implemented. This algorithm is used to give a first estimate of the position of the object in the color image. The range image delivered from a time-of-flight sensor

4.2.8 Status of KE3 Script1: Arranging and interacting with objects

The script stresses the robot's ability to learn from implicit (imitation learning) and explicit (verbal interaction) teaching. The robot learns new skills to manipulate objects and, by so doing, it learns a new task, that of laying out the table. The scenario is depicted in figure 7 is as follows:

The robot watches a human demonstrator laying out a cover on a table, i.e. placing plate, cup, fork, knife, and napkin. The demonstration is repeated several times. Each demonstration is slightly different from the others. While watching the demonstrations, the robot learns the invariants of the task (relative position and orientation of the objects with respect to one another) and new skills such as object-actions relations (how to grip the cup by its handle). Once the demonstrations are finished, the robot reproduces the task.

In the current phase, EPFL and UH implemented methods that 1) extract the important features, i.e. the spatio-temporal correlations across all demonstrations of the task, 2) determine a generic metric to evaluate the robot's imitation performance, and, 3) optimize the robot's reproduction of the task when placed in a new context according to the task metric into the CF-LIF and CF-RG. Successful experiments are reported in part one of the RA4 joint deliverable [7].

4.2.9 Status of KE3 Script 2: Learning to serve a guest

The current state of integration in KE3 Script 2 is shown in a video¹³. Key scenes of the video can be seen in figure 8. The video explicitly shows the integration of the Cogniron functions CF-OR and CF-LCT.

Initially a new object (the yellow chips¹⁴ box) is presented to the robot from multiple views. It captures the object by its range sensor and color camera. The key points of the object are taught to the robot and further range segmented with methods provided by CF-OR. The superquadric model constructed during the object learning process is used to recognize the object in a living room scenario. The ability of the robot to execute grasps on the learned object is also demonstrated.

The next section of the video shows the task learning abilities integrated from WP4.3 into CF-LCT. A first demonstration of the task of serving a guest is given to the system in the task training center at UniKarl. The system acquires the task demonstration through the dedicated sensors and learns the goal spatial relations between the objects in the scene as well as the actions to accomplish this task. From this single demonstration the system is not able to faultlessly execute the task in a scene with relevant changes in the initial setting (note the different object positions of the chips box and the Fanta bottle) because of the collision tainted approach trajectory for a certain object. Once the user gave a second demonstration of the same task, the robot gains the knowledge on the possibility to reorder the subtasks. In the final scene this knowledge gets exploited in order to ensure a successful and faultless task execution. Note that this result is not achieved relying on sub-symbolic path planning methods but on symbolic reasoning on the possible sequence of actions and planning capabilities that select the optimal option.

4.3 Conclusion and Outlook

Building on the results from the first project phase, the implementation, testing and evaluation of the two KE's scripts and its associated Cogniron functions made substantial progress towards the overall demonstration during phase two that can be summarized as follows:

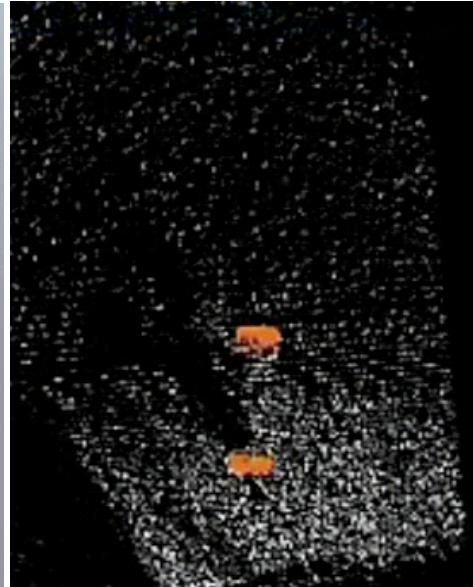
- The extraction of important features of task demonstrations was implemented within CF-LIF on the EPFL platform.
- The reproduction/imitation of recorded gestures was implemented within CF-RG on the EPFL platform.
- Learning and reasoning methods on the overall task knowledge of a robot companion have been implemented within CF-LCT both on the UniKarl training center and the IPAbot.
- Appearance and range image based object modeling was implemented within CR-OR on the IPAbot.
- The inclusion of results from CF-TBP, CF-ACT and CF-GR was started on the IPAbot.
- The progress was shown in the demonstration scenarios of the two scripts "Learning Skills: Arranging and interacting with objects" and "Learning Tasks: Serving a guest".

¹³The video is available on the CD submitted with the deliverables 2005 and can also be downloaded from <http://wwwiaim.ira.uka.de/users/pardowitz/ke3-fin3ivx.avi>

¹⁴or "crisps", depending on the reader's preferences



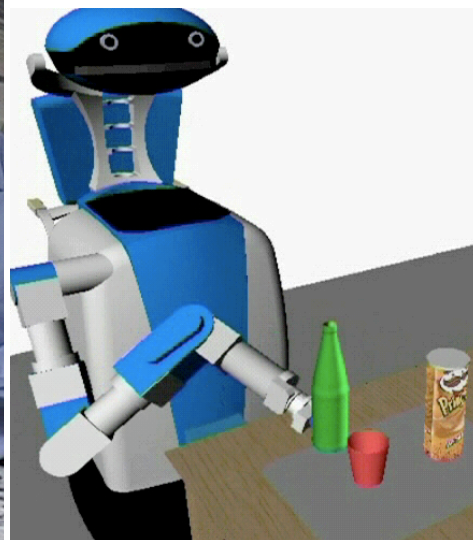
(a) Key points of the object to be learned



(b) Detection of the object in range image



(c) Demonstration of the task to be learned



(d) Successful execution of the task

Figure 8: Key scenes of the Key Experiment 3, script 2 video.

The rest of this section states the work to be done in KE3 during the last two phases. Again, it is organized along the Cogniron Functions involved within KE3. As the main mission of the CFs is to implement the scientific approaches and results gathered in the related RAs, please refer to the according Deliverables and Workplans of RA2, RA4 and RA5 for more detailed information.

CF-LIF will implement the results stemming from the planned considerations of explicit means of training a robot in form of head nods and pointing gestures to help refine and speed up the current purely statistical form of learning in the robot.

CF-RG will include the insights regarding social interaction with respect to the correspondence problem (how to imitate). It will especially implement the methods for the inclusion of self-imitation, scaffolding and sensory-motor experiences into the center of the learning process.

Future work of CF-LCT comprises the inclusion of the results gained in WP4.3 in the last two phases, namely the implementation of reasoning methods for alternative execution possibilities of a task and the extraction of repetitive patterns in a task.

CF-TBP will work on the improvement of existing detection, identification and tracking methods. Advanced initialization methods and plausibility checks will be integrated in the tracking algorithms. The implementation and testing on the robotic platforms will be impelled.

CF-ACT and CF-GR will incorporate final developments on dynamic gestures segmentation and improvements of the classification method, such as HMM, Bayesian Networks and Support Vector Machines (SVMs) on the robot platforms. Additionally, rule-based systems will be introduced to augment the performance of the classification as well as extrinsic features capturing the context of an activity.

5 Software Integration

5.1 Role of Software Integration in Cogniron

Software integration is in the domain of technology transfer between the partners in the Cogniron project. In the parlance of figure 1, it is concerned with the bottom-up link from Research Areas and conceptual integration to the engineering that will produce the running experiments on real robots in realistic settings. This obviously can only be reached in conjunction with the top-down specification methodology already in place since the start of Cogniron. However, an effective exchange platform built around the growing corpus of available implementations will significantly aid in achieving the objectives of the Key Experiments, simply by allowing actual pieces of software to be exchanged in an incremental process that gives feedback on the specifications.

5.2 From Cogniron Functions to Software: Services and Data Structures

According to the technical appendices [1] and the 2005 review in Warsaw [9], there is neither a strict one-to-one mapping between Research Areas (RAs) and Cogniron Functions (CFs), nor is there a very strict responsibility bijection between partners and CFs. Citing the Warsaw review: “[...] *the functions have been chosen [...] in relation to partner interaction rather than for mainly functional or architectural reasons*” (pg. 18).

We argue that for the sake of software exchange, CFs are not an appropriate concept. The notion of atomic *services* has been introduced instead, during the General Assembly (GA) in Amsterdam [11]. Afterward, it has been recognized that more structure is needed for exchanges to take place (GA Karlsruhe [12]), which is presented in section 5.3.

According to Sara Fleury’s presentation [13] in Amsterdam and on the `Service` pages of the Wiki [14], the Cogniron Functions are intended to serve as research objects and do not cover all functionalities required for experimentation. They provide a high-level view for “conceptual” integration. Software integration, on the other hand, requires common interfaces and a way to address technical aspects such as multi-tasking, inter-process communication, task synchronization, and hardware access. Details of such aspects are usually handled by an integration framework and vary among Cogniron partners. The notion of service has been introduced to provide a technical abstraction that should be available in any such framework. A service is made of:

- A set of “pure” algorithmic functions independent of communication framework, operating system, or hardware.
- An API of at least 4 procedures:
 - init** Initialize internal data structures, allocate and check required resources.
 - main** Periodic or one-shot principal activity, the “heart” of the service.
 - end** Clean up and release resources after the main activity has finished.
 - interrupt** A way to cleanly halt the main activity before it has finished.

Each procedure is defined in a functional manner with three parameters:

- input** Semantics, type, and structure of required input information.
- output** Semantics, type, and structure of produced output information (the “result” or “return value” of the procedure, but applicable also to permanent activities).
- report** The status which indicates if the parameters are valid and/or how the procedure has ended (e.g. “OK” or “INTERRUPTED”).

These items constitute the minimum information necessary for exchanging services between partners. More information can be included as needed or available. A catalog of services is being established on the Wiki and presented in the next section. The aim is to allow partners to express needs and provide existing functionality using the service formalism.

Data that will be shared is described on the `DataStructures` page of the Wiki. In particular, this includes parameters for the API. In the long run, XML schemes should be used for these descriptions. However, for the time being any structured format is accepted in order to quickly establish a working base. It has been agreed that the international system of units [41] should be used by default for all applicable parameters, exceptions must be clearly mentioned in the descriptions.

5.3 Enabling Software Exchange: Module Market

The services provide atomic blocks of software functionality and aim at establishing technical interfaces for exchange. In order to facilitate practical software integration it was deemed necessary to define a coarser grained approach: Some services are made to work in conjunction with each other and it does not make sense to distribute them separately; also the requirement to publish precisely defined data structures at the current state of development is not always beneficial, especially if situated at interfaces between two services that can be considered inseparable.

This section presents the `ModuleMarket` page of the Wiki. The purpose of that page is to provide a platform for software exchange by creating a browsable catalog of "functionality" that project partners can use to locate existing software and share requirements. The idea is to use *modules*, *libraries*, and *tools* as the granularity at which things get exchanged. These concepts are illustrated in figure 9.

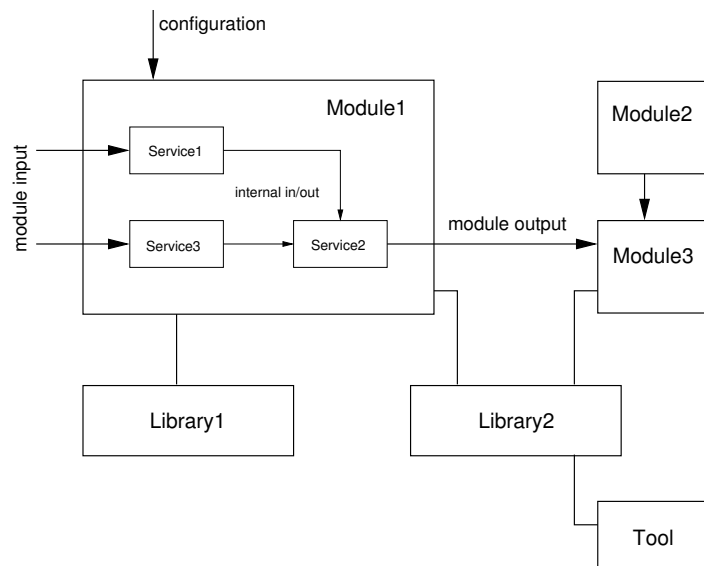


Figure 9: Modules are the units of software that get exchanged on the module market. Services are atomic operations or tasks within a module. Libraries are collections of functionality without a specific associated activity, and tools are things that fit neither of these categories.

5.3.1 Modules

A module is a collection of services with an associated overall (main) functionality (activity). It is not necessary that all modules reside at the same level of abstraction. For example, a GPS module might contain services for configuring and periodically reading a global positioning device; and a module for imitation learning might contain services for extracting features, training a HMM, and periodically interpreting a data stream. A module "belongs" to one Cogniron partner in order to keep software management lightweight inside each module.

A service is an atomic operation inside of a module. Services can be periodic or one-shot, they have strictly defined inputs, outputs, and status reports, as well as initialization, finalization, and run procedures. Inside of their module, the service's in- and outputs are connected in a manner that could be dependent on context (e.g. these pathways would be set by some supervisor). Inputs that come from outside the module, and outputs that have no target within the module, can be considered the containing module's ins and outs.

5.3.2 Tools and Libraries

Things like software frameworks or utility applications are referred to as tools. A library is a "traditional" collection of related functionality. Tools and libraries typically come as a C or C++ package or maybe a collection of related Matlab scripts. The difference between modules and tools/libraries is that the latter don't have an associated main functionality. In other words, a user-space driver would be used as module (for instance by defining file-based in- and output), a collection of image processing functions would be packaged into a library, and a script for translating C structures into XML schemes would be a tool. For practical reasons, a tool/library "belongs" to a single partner, except bridges between technologies coming from different partners. For example, a piece of software that feeds XCF data into an OpenRobots module would belong to UniBi and LAAS.

5.3.3 Other Elements

Apart from a catalogs of available modules, tools, and libraries, the `ModuleMarket` page contains:

Relationship Matrix: To shed light on how partners, Cogniron Functions, Research Areas and so forth are related to each other, in order to clarify roles and responsibilities. This is work in progress, as each partner has different ideas about some of these aspects and interpretations evolve with time; preliminary results are shown in table 1.

Software Release HOWTO: A quick guide to packaging software in order to facilitate the exchange of modules, tools, or libraries. It discusses the question of choosing between binary or source distribution, how to make the package available, and reminds authors to provide at least rudimentary documentation.

Glossary: To explain the terms used on the page.

5.4 Conclusion and Outlook

Several pieces of robotic software technology have to be shared among partners in order to make the key experiments work. We have presented our efforts at structuring and giving content to the exchange platform on the Wiki [14]. The case study in appendix A (p.38) serves as a motivating example and practical guide for future interactions between partners.

- The `Services` and `DataStructures` pages serve to share concrete specifications.
- The `ModuleMarket` page helps to find and suggest exchangeable chunks of software.
- The `KeyExperiments` and `ToolsExchange` pages are for more general collaboration information.
- The `SoftwareIntegration` page serves as pointer to the involved projects and keeps track of actual implementations.

In the upcoming second half of the Cogniron project, our software integration efforts will be devoted to consolidating the growing corpus of shareable software and the associated interfaces and data structures. The Wiki is making good progress towards fostering exchanges that can potentially outlast the remaining duration of the project.

Table 1: Preliminary relationship matrix, assembled from the technical appendices [1], the KE specs [2], the KE questionnaire (see appendix C) of the General Assembly in Amsterdam, as well as existing services, modules, libraries, and tools.

CF	Partners	RA	KE
CF-DLG	UniBi	RA1	KE1
CF-ROR	UniBi	RA2+5	KE1
CF-PTA	UniBi	RA2	KE1
CF-GR	UniKarl	RA2	KE1+3
CF-IA	UH	RA3	KE1+2
CF-LOC	UvA	RA5	KE1
CF-NAV	UvA (LAAS, EPFL)	RA5	KE1
CF-SOC	UH	RA3+6	KE1+2
CF-OR	IPA, LAAS, EPFL, UniKarl	RA5	KE1+2+3
CF-RET	LAAS (EPFL, UH, UniKarl)	RA6	KE2
CF-NHP	LAAS	RA5+6	KE2
CF-MHP	LAAS	RA5+6	KE2
CF-TBP	UniKarl (LAAS)	RA2	KE3
CF-ACT	UniKarl	RA2	KE1+2+3
CF-LCT	UniKarl (EPFL, UH)	RA4	KE3
CF-RG	EPFL, UH	RA4	KE3
CF-LIF	EPFL, UH	RA4	KE3

6 References

6.1 Project Documents and Resources

- [1] Annex 1 *Description of Work*. Cogniron Consortium, 2003.
- [2] D7.1.1 *Specification of the 3 Key-Experiments*. Cogniron Consortium, 2004.
- [3] D7.2.1 *Set-up of the Key Experiment "Robot Home-Tour"*. Cogniron Consortium, 2004.
- [4] D7.3.1 *Set-up of the Key Experiment "Curious Robot"*. Cogniron Consortium, 2004.
- [5] D7.4.1 *Set-up of the Key Experiment "Learning Skills and Tasks"*. Cogniron Consortium, 2004.
- [6] D3.6.1 *Social Behaviour and Embodied Interaction*. Cogniron Consortium, 2005.
- [7] D.4.4.1-2-3 *Report on Successful Decomposition of the Task into Uncorrelated Constraints and Extraction of the Goals, on Studies and Methods for Solving the Correspondence Problem in Context and on Building of and Reasoning over Complex Task Knowledge*. Cogniron Consortium, 2005.
- [8] D6.2005 Joint RA6 Deliverable *Decision, Intentionality and Initiative for a Robot Assistant*. Cogniron Consortium, 2005.
- [9] Review Report N°1, Covering period 01.01.2004 - 31.12.2004
- [10] Periodic Activity Report. Cogniron Consortium, 2004.
- [11] General Assembly Meeting, Amsterdam June 05.
http://www.cogniron.org/internal/iMeetings_AMS05.php
- [12] General Assembly Meeting, Karlsruhe November 05.
http://www.cogniron.org/internal/iMeetings_KA05.php
- [13] Sara Fleury LAAS, slides of *Cogniron Services*. Presented during the General Assembly Meeting, Amsterdam June 05.
- [14] Cogniron Wiki collaboration platform <http://www.cogniron.org/wiki/>

6.2 Applicable Documents

- [15] J. Fritsch, M. Kleinhagenbrock, A. Haasch, S. Wrede, and G. Sagerer. A flexible infrastructure for the development of a robot companion with extensible HRI-capabilities. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3419–3425, Barcelona, Spain, April 2005.
- [16] J. Fritsch, B. Wrede, and G. Sagerer. Bringing it all together: Integration to study embodied interaction with a robot companion. In *AISB 2005 Symposium – Robot Companions: Hard Problems and Open Challenges in Robot-Human Interaction*, Hatfield, England, April 2005.
- [17] E. Lopez-Damian, D. Sidobre & R. Alami. A grasp planner based on inertial properties. In *ICRA 2005 — Proc. IEEE International Conference on Robotics and Automation*, pages 766–771, Barcelona, April 2005.

- [18] E. Lopez-Damian, D. Sidobre & R. Alami. Grasp Planning for Non-Convex Objects. In *ISR 2005 – Proc. 36th International Symposium on Robotics, Tokyo, Japan, November 2005, Best paper award of the symposium.*
- [19] N. Do Huu, W. Paquier, R. Chatila. Combining Structural Descriptions and Image-based Representations for Image, Object, and Scene Recognition. In *IJCAI 2005 — Proc. of 19th International Joint Conference on Artificial Intelligence (IJCAI 2005), August 2005.*
- [20] I. Herrera Aguilar, D. Sidobre. On-line trajectory planning of robot manipulator's end effector in Cartesian space using quaternions. In *ISMCR 2005 — Proc of 15th International Symposium on Measurement & Control in Robotics, Brussels (Belgium), November 2005*
- [21] A. Billard, S. Calinon and F. Guenter: Discriminative and Adaptive Imitation in Uni-Manual and Bi-Manual Tasks. In *Robotics & Autonomous Systems, special issue on the social mechanisms of robot programming by demonstration (In press).*
- [22] S. Calinon, F. Guenter and A. Billard: Goal-Directed Imitation in a Humanoid Robot. In *Proceedings of the International Conference on Robotics and Automation, ICRA'05, Barcelona, Spain, 18-21 April 2005.*
- [23] A. Alissandrakis, C. L. Nehaniv, K. Dautenhahn, and J. Saunders: Imitating using JABBER-WOCKY to achieve corresponding effects in context. Computer Science Technical Report 441, University of Hertfordshire, Hat eld, Hertfordshire, UK. Submitted to *IEEE Trans. Syst., Man, Cybern. Part A: Systems and Humans*, 2005.
- [24] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn: Correspondence mapping induced state and action metrics for robotic imitation. Computer Science Technical Report 440, University of Hertfordshire, Hat eld, Hertfordshire, UK. Submitted to *IEEE Trans. Syst., Man, Cybern. Part B: Cybernetics*, 2005.
- [25] M. Pardowitz, R. Zöllner, S. Knoop, and R. Dillmann: Incremental learning of tasks from user demonstrations, past experiences and vocal comments. Submitted for the *IEEE Trans. on Systems, Man, and Cybernetics Special Issue on Robot Learning by Observation, Demonstration and Imitation*, 2006.
- [26] M. Pardowitz, R. Zöllner, and R. Dillmann. Learning sequential constraints of tasks from user demonstrations. In *Proc. IEEE-RAS Intl. Conf. on Humanoid Robots (HUMANOIDS05)*, pages 424-429, Dec 2005.
- [27] S. Knoop, S. Vacek and R. Dillmann: Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2006)*, May 15, Orlando, Florida. (accepted for publication).
- [28] S. Knoop, S. Vacek, R. Dillmann, S. Brännström, H. I. Christensen: Extraction, Evaluation, Selection and Classification of Motion Features for Human Activity Recognition, Internal Report, Universität Karlsruhe, 2005.
- [29] J. Kubacki, B. Giesler, C. Parlitz: Active Autonomous Object Modeling for Recognition and Manipulation: Towards a Unified Object Model and Learning Cycle. *Autonome mobile Systeme*, Stuttgart, Germany, 2005.

6.3 Reference Documents and Resources

- [30] Alexandre Lampe and Raja Chatila. *Performance Measure For The Evaluation of Mobile Robot Autonomy*. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2006), May 15, Orlando, Florida. (accepted for publication).
- [31] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, March 1997.
- [32] J.-C. Latombe. *Robot motion planning*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1991.
- [33] Roland Philippsen. *Motion Planning and Obstacle Avoidance for Mobile Robots in Highly Cluttered Dynamic Environments*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2004.
- [34] Sean Quinlan. *Real-Time Modification of Collision-Free Paths*. PhD thesis, Computer Science Department, Stanford University, 1994.
- [35] Roland Siegwart, Kai O. Arras, Samir Bouabdallah, Daniel Burnier, Gilles Froidevaux, Xavier Greppin, Björn Jensen, Antoine Lorotte, Laetitia Mayor, Mathieu Meisser, Roland Philippsen, Ralph Pigué, Guy Ramel, Gregoire Terrien, and Nicola Tomatis. Robox at Expo.02: A large-scale installation of personal robots. *Robotics and Autonomous Systems*, 42:203–222, 2003.
- [36] M. Isard, and A. Blake. CONDENSATION – Conditional Density Propagation For Visual Tracking. In *IJCV 1998 – International Journal on Computer Vision*, Hingham, USA, 1998.
- [37] M. Isard, and A. Blake. ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework. In *ECCV 1998 – European Conference on Computer Vision: Proceedings of the 5th European Conference On Computer Vision-Volume I*, London, UK, 1998.
- [38] M.K. Pitt, and N. Shephard. Filtering Via Simulation: Auxiliary Particle Filters. In *JASA 1999 – Journal of the American Statistical Association*, 1999.
- [39] P. Torma, and C. Szepesvári Sequential Importance Sampling For Visual Tracking Reconsidered. In *AIS 2003 – AI and Statistics*, 2003.
- [40] J. Minguéz, L. Montano. Nearness Diagram Navigation (ND): Collision Avoidance in Troublesome Scenarios. In *IEEE Transactions on Robotics and Automation*, p 154,2004.
- [41] The NIST Reference on Constants, Units, and Uncertainty.
<http://physics.nist.gov/cuu/Units/>
- [42] Columbia Object Image Library.
<http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html>
- [43] PerMIS'04 *Performance Metrics for Intelligent Systems Workshop*
http://www.isd.mel.nist.gov/PerMIS_2004/

A Account of Technology Transfer Between UniBi and LAAS

This appendix is an account of the initial technology transfer between Uni Bielefeld and LAAS, which took place in November 2005 (three days on-site in Bielefeld, after coordination and preparation via email and telephone).

A.1 KE1 Software Integration Context

Several partners contribute software to KE1 “Robot Home Tour”. UniBi is the KE leader and provides the hardware as well as the majority of software. At the current state of development, software from LAAS and EPFL has also been integrated, and contributions from UvA are expected to follow. No common framework has been chosen for these integrations, thus several components come into play for KE1 software integration. They are presented in table 2.

At the time of initial technology transfer between LAAS and UniBi, Biron’s state of development had reached a point where it was necessary to integrate localization and navigation (path planning and obstacle avoidance). The fact that neither CF-LOC nor CF-NAV had reached sufficient maturity created a issue for testing the data structures defined for that purpose in the KE1 architecture. It was thus chosen to use existing and/or “mock” implementations in order to

1. Validate the KE1 architecture.
2. Exchange tools for integration between UniBi and LAAS.
3. Gather experience on the challenges of technology transfer.

Table 2: Software tools and libraries involved in KE1 at the time of writing

Tool	Partners	Remarks
XCF	developed at UniBi	XCF is an XML-based communication framework used on the mobile robot Biron on which KE1 is being developed in C++ under Linux. Communication is stream oriented (polymorphic object serialization).
Player/Stage	used at UniBi and UvA	Player/Stage is an open-source network abstraction for mobile robots which has gained some popularity. Communication is stream oriented (simple data), bindings exist for a large range of languages.
OpenRobots	developed at LAAS	OpenRobots is a collection of portable mobile robot libraries and control modules. It is based on a collection of C libraries and a code generator used on all robots there, it also runs under Linux. It provides multitasking abstraction, communication is based on shared-memory.
Sunflower	developed at EPFL	Sunflower is a C++ mobile robot library providing C bindings to a subset of its functionality. Originally aimed at providing an abstract framework, it now mainly contains obstacle avoidance and a simple path planner. Communication is based on function call semantics.

A.2 Objectives of Technology Transfer

This section explains *what* got exchanged, and *why* we concentrated on these aspects. Generally speaking, it was necessary to create incentives for both involved partners to invest work.

Install OpenRobots on Biron : UniBi gains potential for using LAAS modules, as these are developed within OpenRobots and can then be ported to other robots with very little effort. LAAS gains more users for their OpenRobots, as well as feedback about possible installation and documentation problems.

Communication between XCF and OpenRobots : Both gain insights into the other's technology, allowing to better assess the costs and benefits of implementing a generic bridge between the two frameworks.

Navigation and “fake” localization for Biron : UniBi gains functionality for KE1 development, in the sense of concept checking as well as real-world testing: Path planning and obstacle avoidance, data flow between dialog and robot control. LAAS gains know-how for integrating CF-DLG on their robots in the future.

A.3 Preparation and On-Site Work

The overview diagram in figure 10 shows which parts of the KE1 architecture were directly concerned by this first step. The idea is to replace LOC by a very simple module that maps location names to coordinates using a-priori information from a file, which can then be used by NAV to internally direct obstacle avoidance using Cartesian coordinates relative to the robot frame. The call graph in figure 11 clarifies the information and control flow. The three subsystems (NAV, LOC, ObstacleAvoid) run as OpenRobots processes on a dedicated computer and communicate with the rest of the system via TCP/IP, as shown in the development setup diagram (figure 12).

In order to bridge the technological gap between XCF (runtime polymorphism, object- and stream-oriented, XML standards) and OpenRobots (most things fixed at compile time for efficient shared-memory access), it was decided to use the least common denominator technology: Pure C interfaces implemented in small standalone wrapper libraries.

A.4 Results

Two notable problems arose before Biron finally navigated autonomously after receiving a vocal command such as “Biron, go to the kitchen”. This demo can be viewed on video¹⁵.

Unforeseen crosstalk : There seemed to be a multi-threading problem when running both OpenRobots and XCF on a host. This problem crept up rather late one evening in conjunction with other issues, and it was decided to decrease the number of potential problem sources by writing a standalone application that wraps libsunflower instead of using the OpenRobots module. Thus it was guaranteed that the XCF client was executing in a single thread.

This measure solved the issue, but later analysis leads us to believe that the source of the problem is not due to multitasking, as had seemed to be the case during debugging. This remains an open question.

¹⁵Either on the CD submitted with this deliverable, or on <http://www.laas.fr/~rolo/media.html>.

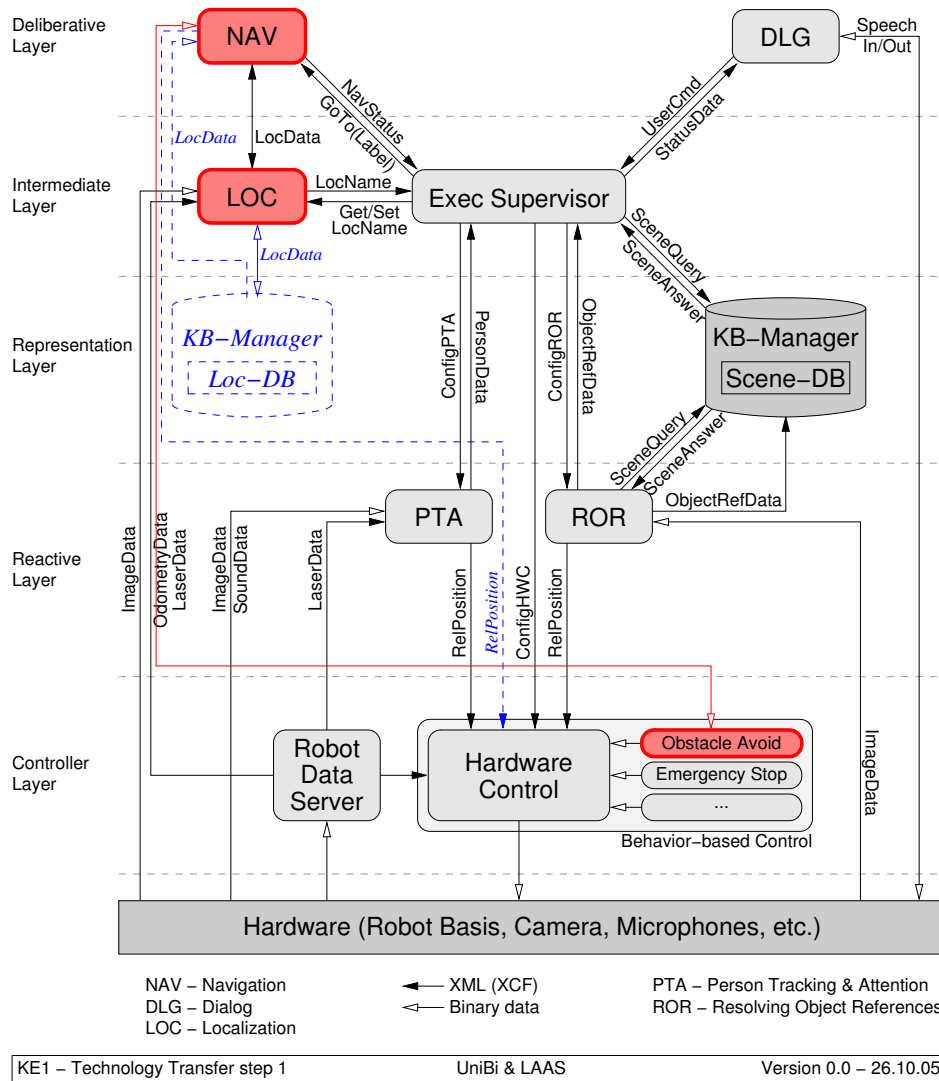


Figure 10: Modified KE1 architecture for reducing the number of interfaces to be taken into account during on-site work, with processes running under OpenRobots marked in thick red (NAV, LOC, ObstacleAvoid, link from NAV to ObstacleAvoid), and dashed blue indicating which components were temporarily removed (KB-Manager, Loc-DB, LocData, RelPosition). Compare with the overall KE1 architecture in figure 2 (p.13).

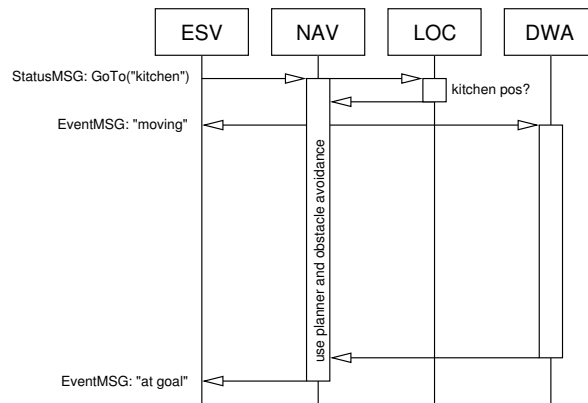


Figure 11: Typical information flow for integrating path planning and obstacle avoidance into the existing KE1 system. ESV denotes the execution supervisor that routes the “GoTo” command from dialog to NAV, which queries the mock LOC about the position that corresponds to the given label. Then, NAV informs ESV that the robot is moving and uses DWA (path planning and obstacle avoidance) to control motion commands from laser data and pose information. NAV informs ESV when the goal is reached. Not depicted in this diagram are: (1) the periodic messages from NAV to ESV that serve as a watchdog, and (2) the link from DWA to motor control, which was implemented on-site using XCF streams.

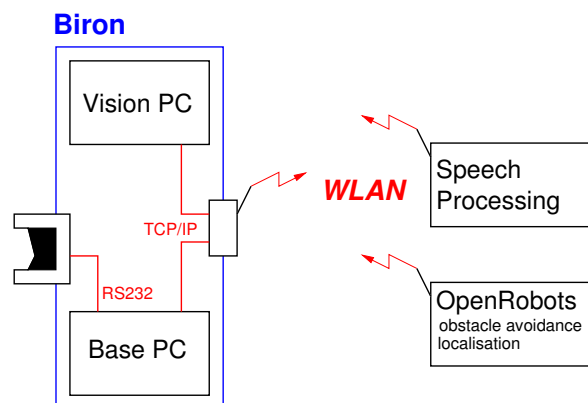


Figure 12: Physical development setup for on-site work. It was decided to use wireless communication even for the obstacle-avoidance control loop in order to keep potential crosstalk between subsystems as small as possible.

Wireless Networking : During development on the cabled network, XCF communication worked fine, but switching to a wireless setup disrupted communication. This was probably due to an unforeseen interaction between a newly installed wireless bridge and the wireless router. Again for saving developer time, it was decided to transfer the NAV subsystem to the dialog laptop which was known to have functional networking.

This decision made it possible to demonstrate a functioning integration, however the already scarce computing resources of the laptop were depleted and substantially lowered the performance of the obstacle avoidance.

A.5 Conclusion

Three pragmatic rules can be formulated from the first software exchange between UniBi and LAAS, to be applied to the wrapper libraries that act as glue between technologies:

- Do not use polymorphic APIs, they are not easily expressed in procedural programming languages.
- Adhere to coding discipline by using “best practices” (constants, error codes), avoid advanced language features such as exceptions.
- Use the least common denominator technology and keep interfaces as simple as possible:
 - standard C prototypes (you can still *implement* in C++)
 - use C builtin data and array types

B Detailed Integration Account for KE2

B.1 Objectives and Organization

The objective of KE2 relates to a curiosity behaviour triggered by a robot drive for learning more about the environment, but it also includes issues related to the capacity for a robot to take initiatives. The underlying scientific issues addressed in the Research Areas are the following:

- Object learning and recognition (RA5);
- Interpretation of human activities and posture (RA2);
- Close interaction with humans (RA3);
- Intention attribution (robot intention expression to be legible to the humans) (RA6);
- Situation understanding, decision making, collaborative problem solving (RA6);
- cognitive architectures (RA6).

Two separate scripts are developed, one emphasizing robot curiosity, and the other robot initiative and intentionality.

The first script describes how a robot builds a model of a newly discovered object through sensing and manipulation. The second script describes how a robot interprets a situation (involving a human) that requires its intervention for fetching an object and handing it to the human.

For the last period, the main focus was on setting up the necessary material for demonstrating the scenarios on the one hand, and integrating parts of the software developed in the various RA's involved. In order to allow simultaneous work and proceed toward partial integration without waiting for the target mobile manipulator to be fully functional, we have chosen to build partial experiments on three different platforms:

- Experiment KE2-1: Object modeling and handling by a mobile manipulator
- Experiment KE2-2: Toward human-friendly Navigation
- Experiment KE2-3: Curiosity and learning

Each experiment corresponds a partial integration of the targeted functions involved in the global KE2 scenario. Besides, the three platforms share the same underlying software architecture and tools¹⁶.

B.2 Experiment KE2-1: Object Modeling and Handling by a Mobile Manipulator

B.2.1 Objectives and Context

Experiment KE2-1 has been designed to implement generic object modeling and manipulation abilities that are planned to be used in the framework of KE2.

The target robot is a Neobotix platform equipped with a PA10 6-DOF arm (Fig 13). It has been purchased in February 2005 and has been equipped with a home made gripper, two stereo camera systems and a 6 DOF force sensor (see section B.2.2).

¹⁶Released under a BSD open source license on <http://softs.laas.fr/openrobots/>

The scenario for the experiment (see section B.2.3) has been refined and instantiated into a set of Cogniron Services that are intended to implement the Cogniron Functions to model the object from stereo vision (see section 1) and to plan its handling (see section 2).

All the Cogniron and Supporting services required to achieve this scenario have been detailed and organized within a software architecture presented in paragraph section B.2.4.

As soon as the hardware platform was ready we started the integration of the various services (see section 3). In December 2005 all of them have been at least tested individually on the real platform (see section 3). The next phase will be debug the overall integration and to demonstrate the whole scenario.

B.2.2 Robot Platform

Jido is a MP-L655 platform from Neobotix, equipped with a Mitsubishi PA-10 arm (with 6 degrees of freedom). Several sensors are available on the platform: sonars, two Sick laser range finders, two stereo camera banks (one mounted on the arm and the other on a pan-tilt unit on the base platform) and several contact sensors on the gripper. Two on-board computers (Intel Pentium 4 processors, at 3GHz, with 512 megabytes of RAM each), using the Linux operating system, are used to run all software presented in section B.2.4.

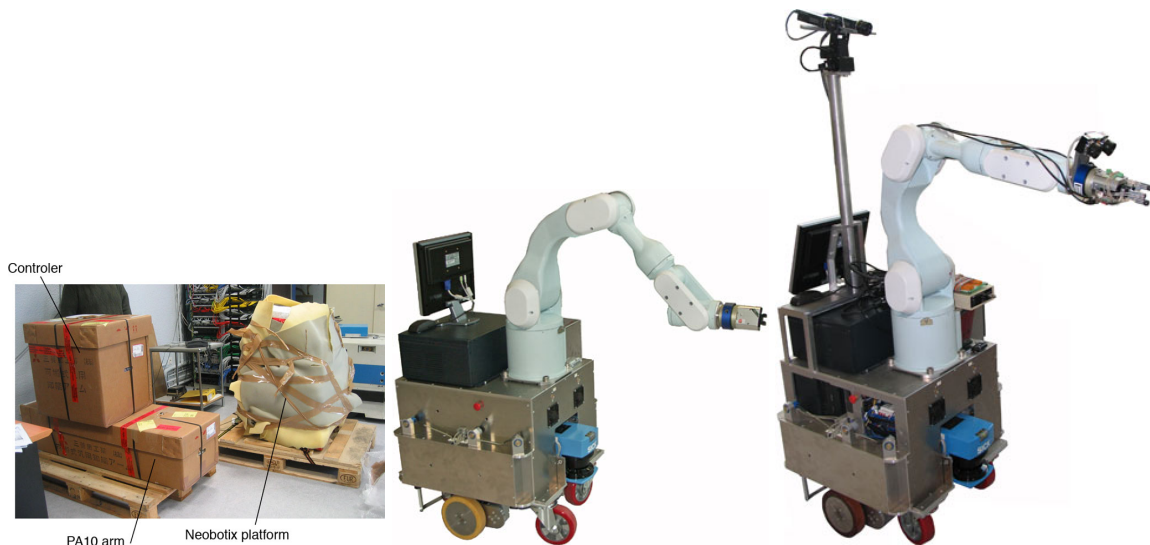


Figure 13: Jido from February to December 2005

The original two parallel jaw gripper was equipped with two fingers providing three contact points. Jaw opening potentiometer, contact force sensor resistors and strain gage to measure applied forces have been installed and connected to a micro-controller.

B.2.3 Scenario

As already mentioned, this scenario is intended to provide a set of robust and generic object manipulation abilities to be used in the global KE2 demonstration on a mobile manipulator.

The robot (a mobile manipulator) detects an object on a table. It must first model it moving to various viewpoints the stereo system mounted on the arm. The computed 3D model of the object is memorized by the robot. From this model the robot is able to identify it later on, to locate it and possibly to grasp it.

It thus plans a path to catch the object with its gripper without colliding the table or others obstacles around. The object is grasped. It can then be given to a person or placed elsewhere. In this first experiment, the object is assumed to be isolated on the table. The robot will also be able to segment the table plane.

B.2.4 Software Architecture

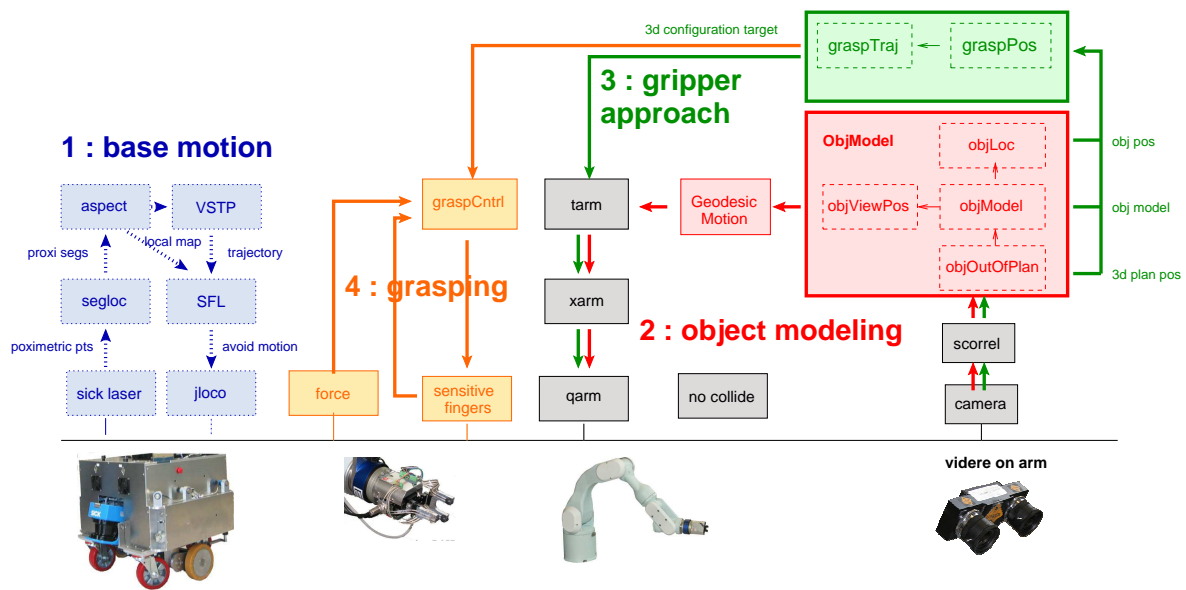


Figure 14: Current software architecture with all the services involved

The whole scenario described in section B.2.3 requires 23 services organized in four main functions as described in figure 14:

1. base motion (blue)
2. object modeling (red)
3. gripper approach (green)
4. grasping (orange)

The arrows represent the data flow between the services. The execution of the services is coordinated by a Tcl script not represented in the figure (later on the supervisor that will be first implemented on Rackham will be integrated on Jido).

The execution sequence is the following:

1. First the robot reaches the table using the **base motion** function: with a sick laser (*sick*) it localizes itself (*segloc*) and builds a local map (*aspect*) in which a path is computed (*VSTP*) to reach a final configuration near the table without hurting obstacles. Finally, it executes this plan thanks to *SFL* local avoidance and *jloco* speed servo-controller. This function is still in dotted line because for now it has been tested only on the Bielefeld platform. However the implementation on Jido, which was not first priority for this demonstration, is nearly completed.

2. Once near the table it executes the **object modeling** loop: from a given position it acquires stereo-images (`camera`), computes a 3D image of points (`scorrel`), initializes or updates the 3D model of the table and of the object (`objOutOfPlan` and `objModel`), chooses a new view point (`objViewPos`) and moves its arm toward this new position (`GeodesicMotion`, `tarm`, `xarm` and `qarm`). This step is re-iterated until a sufficient object model is obtained.
3. From the 3D object model the robot is able to identify it and to localize it (`objLoc`) relatively to its gripper. The **gripper approach** loop can be executed: the system is able to compute a grasping position of the object (`graspPos`) and a trajectory to reach this position without collision (`graspTraj`). This step can also be re-iterated until the object is well positioned between the fingers.
4. The final function consists in **grasping** the object using the various sensors on the fingers (`fingers`) and the 6D force sensor.

Two of these functions, "object modeling" and "gripper approach" (in bold in the figure) are related to Cogniron Functions and described in section 1 and section 2. The other ones are Supporting Services and are presented in section 3.

B.2.5 Implemented Services

The three following sub-sections present the main services that have been developed and implemented on the robot for the demonstration.

The first one `Mobj` allows to model objects on a table from stereo-vision in order to be able to manipulate them.

The second one `GraspPlan` allows to plan a collision-free path to take and manipulate the object.

The last sub-section will describe shortly the main support services: the services that are mandatory to accomplish the task but that are not tackled by Cogniron Functions.

1. Service "Object Modeling"

This service implements the function required to build the geometrical model of an object. This 3D model will be used by other services, to find optimal grasping positions depending on the object and the gripper shapes, and then to grasp and manipulate the object.

At this preliminary step of the KE2 experiment, we focus only on the construction of a 3D model for an isolated, rigid and static object, put on a horizontal plane (typically, a table or the ground). The model is built from sensory data acquired only by the stereo-vision sensor mounted on the Jido arm. By now, the sensor is moved around the object using the arm, considering that the mobile robot is docked along the table.

Purposes:

The modeling function is an incremental process. A calibration procedure must be performed off line before any experiment in order to estimate intrinsic parameters of the stereo sensor and the hand-eye transform T_{eff}^{stereo} (situation of the sensor frame w.r.t. the effector frame).

Several algorithms are integrated to build the object model:

- 3D data acquisition from images acquired by the stereo sensor. A dense stereo-vision method provides a 3D image, related to the left intensity image. For every matched pixel in this left image, a 3D point is provided by triangulation, given the parameters of the stereo sensor.

- extraction of the object points from the 3D image, using contextual information. Here it is supposed that the object is on a horizontal plane. A simple planar segmentation (from a clustering approach or a 3D Hough transform) allows to extract and to remove points located on the the supporting plane. More dedicated methods using intensity or color attributes could be also executed for the same objective.
- estimation of the sensor position, called T_{ref}^{stereo} with respect to a static local reference frame, called hereafter \mathcal{R}_{ref} , typically the docking position of Jido close to the table. The T_{ref}^{stereo} transform is computed using the hand-eye transform T_{eff}^{stereo} , the current effector position T_{arm}^{eff} computed from encoder data using the geometrical model of the arm, the arm position on Jido T_{jido}^{arm} given by mechanical measurements, and finally if required, by the Jido current position with respect to the reference frame T_{ref}^{jido} , computed from odometry and from a localization module.
- registration of all sensory data. Using the estimated sensor positions, all stereo data can be fused in the same cloud of 3D points, expressed in \mathcal{R}_{ref} , using a classical ICP algorithm (for *Iterative Closest Point*). Several operations (decimation, 3D edge extraction. . .) allows to limit the ICP complexity.
- construction of a triangular mesh from a cloud of 3D points. By now, it is based on the *Marching Cubes* algorithm.

Relationships with RA and CF:

This service is a partial instance of the CF-OR (*Object Recognition and Modeling*) Cogniron function: object recognition and localization procedures are not integrated by now. Stereo-vision allows to obtain directly the fusion between intensity, color and 3D sensory data; so appearance-based patterns (typically interest points or SIFT patches) will be easily located on the geometrical model of the object, so that object recognition and localization will be based on such patterns in the next step.

This service is related to the work package WP5.2 devoted to the perception on objects in RA5. In KE2 we aim at integrating and comparing two different approaches:

- UniKarl and IPA are working together on a combined representation, acquired using a Swiss Ranger sensor coupled with a color camera; UniKarl deals with the construction of a 3D geometrical model, using color to segment the object from the background, while IPA is concerned by object recognition from appearance-based representations from SVM classifiers.
- LAAS will improve the incremental modeling method described above, and will integrate a Bayesian approach to cope with active recognition of an object.

Integration:

The system has been integrated on the robot Jido using a Videre stereo sensor (cameras 1/3", 9cm baseline equipped with 4.8mm lenses) mounted close to the end effector of the robot's arm. Like it is shown figure 13, another stereo sensor (cameras 1/2", 20cm baseline equipped with 3.6mm lenses), are mounted on a mast behind the arm: it will be more devoted to visual navigation (landmark detection and recognition, visual SLAM...).

The stereo sensors are calibrated using a planar calibration tile (LAAS method, similar to operators available in OpenCV or Matlab). The hand-eye calibration is solved using a method inspired from the Dornaika/Horaud one.

First results are presented on Fig 17, for polyhedral objects (a simple box, with enough textures to allow stereo matchings) or for more complex ones (eraser, camera stand, tube...). The built triangular meshes will be transmitted to the "Grasp Planning" service.

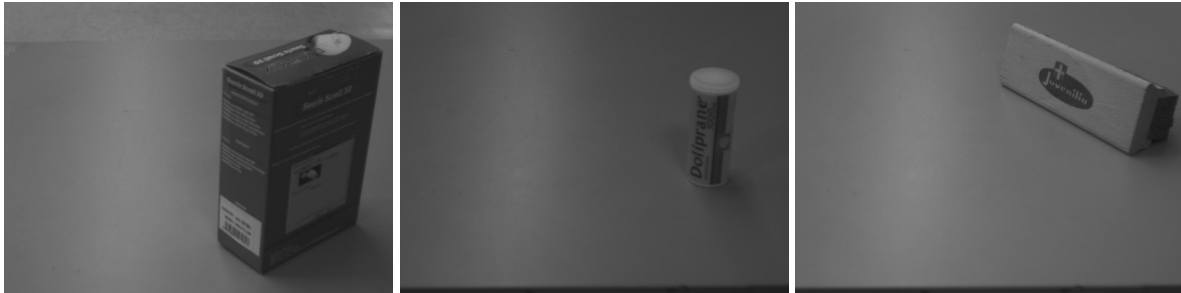


Figure 15: Several objects to be modeled and grasped: a box, a tube, an eraser

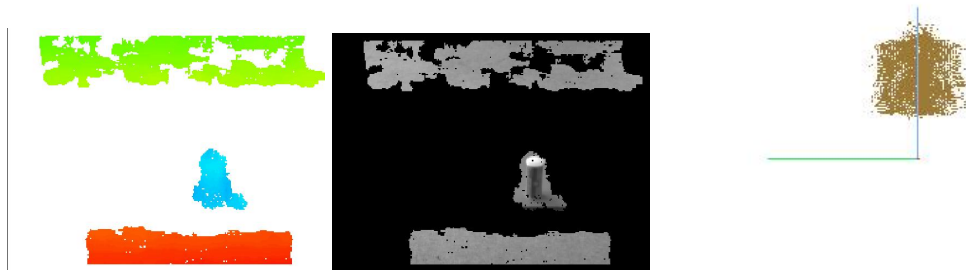


Figure 16: (Modeling of the tube: (left) disparity image, (middle) matched pixels, (right) cloud of points after the fusion of 8 3D images

Next step:

many extensions are foreseen for this service:

- on line calibration of the hand-eye transform using a EKF-based approach, dealing also with the object localization from interest points;
- improvements of the modeling functions, evaluating different strategies for the ICP procedure (incremental or global...) and different algorithms to build the triangular mesh (*Marching Cubes, Ball Pivoting ...*).
- evaluation of an existing Next Best View method, in order to minimize the number of views required to build a complete object model.
- integration of data acquired from the two stereo sensors, and using several docking positions of Jido around the table.
- integration in the same object representation, of appearance-based patterns suitable to recognize and localize the object from any viewpoint.
- other object/sensor configurations (for example, the object is moved by an operator or by the arm, in front of a stereo sensor).

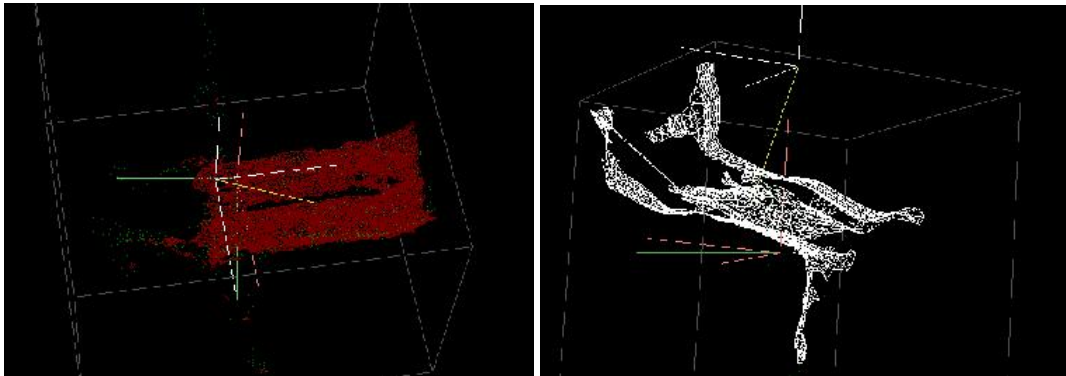


Figure 17: (right) ICP result on 5 successive 3D images of the box; (left) the corresponding triangular mesh

2. Service “Grasp Planning”

Purpose:

Based on triangular mesh model of object, robot and environment, the objective is to compute a plan to grasp an object and perform simple tasks like pick and place or hand over. The two cases of hand over task (with a human or with a second robot) was studied. The grasp planner is based on inertial properties and object decomposition.

Relationships with RA and CF:

This grasp planner is part of the general motion and task planning abilities that a robot companion should possess. Consequently it is linked to RA6 and RA6 and to the CF called CF-MHP (*Manipulation in Human Presence*).

Integration:

The system was integrated on the Jido robot as an OpenRobots module. An important programming effort has been dedicated to embed a number of software libraries (Move3D tools).

References

Results were presented in Barcelona IEEE-ICRA 2005 [17] and at the International Symposium on Robotics ISR 2005 [18] held in Tokyo where the paper obtained the best paper award of the symposium.

Results:

The work was mainly done in simulation both on virtual models and real models publicly downloadable from the Web.

Figures 18 and 19 show illustrative results. The influence of obstacles on the grasp position is illustrated for a bottle. The grasping of the mug by the handle demonstrates the efficiency of the convex decomposition approach. Supplementary videos¹⁷ are also available.

The hand over task is illustrated for a bottle and a glass. Table 3 presents computing time for grasp planner obtained using a 500 MHz Solaris SunBlade. The first three rows are for Mitsubishi arm and a convex object decomposition. The last two rows(†) are for a hand over task involving a second robot with a Gt6a arm, the object decomposition strategy differs from

¹⁷Videos are on the CD submitted with the deliverables 2005 (section RA7).

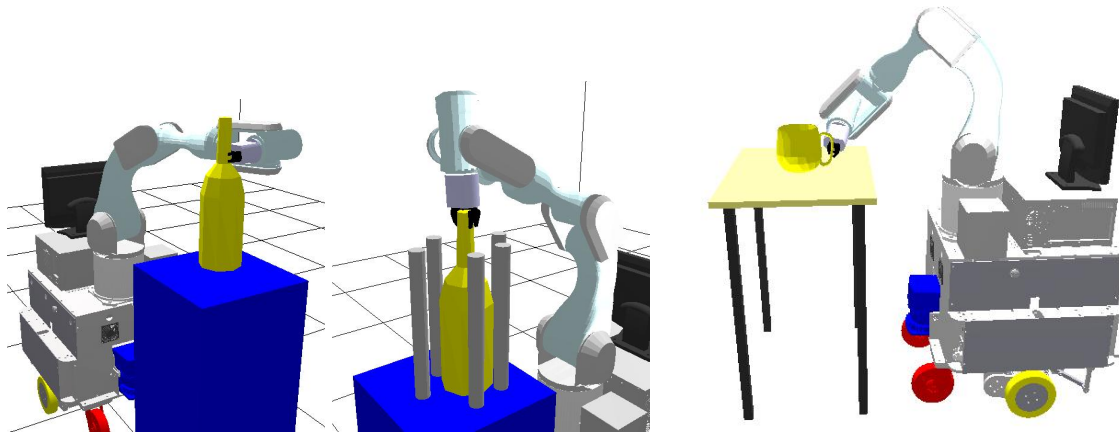


Figure 18: Two different solutions for grasps a bottle; right: grasp of a mug by the handle

convex decomposition in the cutting phase. Three cuts on the object through the inertial axes are made in the latter case.

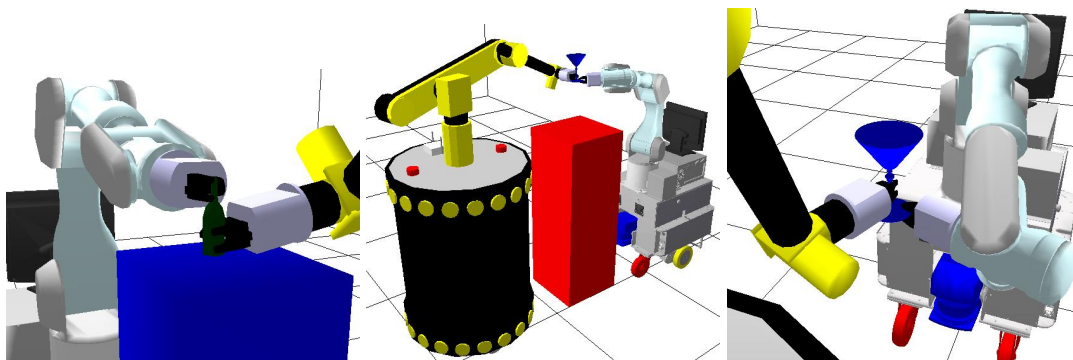


Figure 19: Hand over of a bottle and a glass

Note that the planner is able to take into account a double-grasp when computing its own grasp configuration.

Next step:

Firstly we have to test the system on real manipulations.

For the planner, we are going to extend it to manage more complex grippers and to operate in more complex environment. A canonical task would be to clear a table where many objects have been placed. Other manipulation techniques will be studied like pushing objects. For example, pushing an object before grasping it.

3. Supporting Services

Arm Control with `qarm`, `xarm` and `collide`:

`Qarm` is a OpenRobots module developed to interface micro-controllers of the Mitsubishi PA10-6C arm at the joint level. `Qarm` is associated with the `collide` module to detect collision between parts of the arm and the mobile platform.

Table 3: Grasp planner computing time (seconds)

Object	Cut Time	Grasp Time	Total Time	Facets
Bottle	0.12	36.7	36.82	80
Mug	0.71	34	34.71	499
Glass	13.58	135.56	149.14	2750
Bottle†	6.11	42.86	48.97	80
Glass†	21.54	301.2	322.74	2750

Xarm integrates an on-line soft motion trajectory planner to facilitate definition of sensory-motor loop in Cartesian space using qarm module. This work was presented at the 15th International Symposium on Measurement and Control in Robotics, 08-10 November 2005, Brussels, BELGIUM [20]. A video¹⁸ showing a trajectory computed by the grasp planner and executed by Jido is available.

Hand Control with fingers, force and graspCntrl:

Fingers and force module respectively interface fingers sensors (Force Sensor Resistors and strain gages for contact forces, finger position) and a six-axis force sensor. The graspCntrl module that must control gripper is not yet entirely implemented. Its two main functions are to control grasping and passing an object to a person.

Base Localization with sick, segloc and pom:

We are currently integrating the same modules as for Rackham (see section 3).

Motion and Navigation with JLOCO and SFL:

The JLOCO service provides basic motor control and odometry for the Neobotix platform. It communicates with the motors and the gyro of Jido through the CAN bus, using a CAN driver developed at LAAS.

The SFL service wraps the “Sunflower” mobile robot library from EPFL. It provides path planning and obstacle avoidance that has been developed for the Expo02 event in Switzerland. It is also used on Biron for KE1.

B.2.6 Results and Conclusions

Work done in 2005:

- Equipment and a new mobile manipulator
- Installation of software integration tools.
- Definition of a set of Cogniron services (and their interaction) related to object modeling and manipulation
- Adaptation of a set of supporting services (localization and motion)

¹⁸Video is on the CD submitted with the deliverables 2005 (section RA7).

- Development and Integration of new supporting software
- Development of a grasp planner
- Development of a new control law for manipulation
- Development of a stereo-based object modeling service

All essential ingredients are now available to build and demonstrate an autonomous and generic ability to model and handle objects.

B.3 Experiment KE2-2: Towards Human Friendly Navigation

B.3.1 Objectives and Context

Experiment KE2-2 has been designed to investigate and illustrate a mobile robot abilities to navigate in the presence of humans and to approach persons in order to interact or to hand objects.

The robot must be able to detect humans presence in its vicinity and to interpret their trajectories, in order to secure their displacements —by avoiding them or to getting out of their way— or to interact with them.

One contribution focuses on the development of people detection and tracking modalities from the video stream of the on-board color camera, each modality being suited to a specific H/R interaction distance. The other contribution focuses on the illustration of a Human Aware Motion Planner.

B.3.2 Robot Platform

Rackham is a B21r robot (iRobot). It is a 4-feet (52 cm) wide and 20-inches (118 cm) tall cylinder topped with a mast supporting a kind of helmet. It integrates 2 PCs (one mono-CPU and one bi-CPUs running P3 at 850 MHz). We have extended the standard equipment with a pan-tilt Sony camera EVI-D70 attached under the helmet, a digital camera mounted on a Directed Perception pan-tilt unit, an ELO touch screen, a pair of loudspeakers, an optical fiber gyroscope and wireless Ethernet.

B.3.3 Scenario

A environment model is assumed to have been acquired previously. Such a model allows the robot to localize itself and to synthesize paths that avoid fixed static obstacles (walls, furniture).

Rackham is motionless, and looks at the room entrance. It fires its long-range *monitoring modality*, so as to detect and track any person entering the area.

A person comes in. Her aim may be to go and sit down on a chair located behind a table. When the person gets closer to the robot, the latter switches to the mid-range *human tracking modality*, which is more suited to the associated relative H/R distance. During this second phase, the robot estimates the relative position of the targeted human, and rotates in order to keep him within sight. No active vision strategy is involved at the moment.

Based on appropriate heuristics, still running the *mid-range human tracking modality*, the robot detects that the human sits down.

Thanks to its *Navigation in Human Presence* functionality, the robot plans a trajectory from its current location to a zone —a priori defined in the environment map— in front of the chair. The execution of this human-safe and human-comfortable trajectory is then performed in a reactive manner, so as to avoid any passer-by sharing the environment.

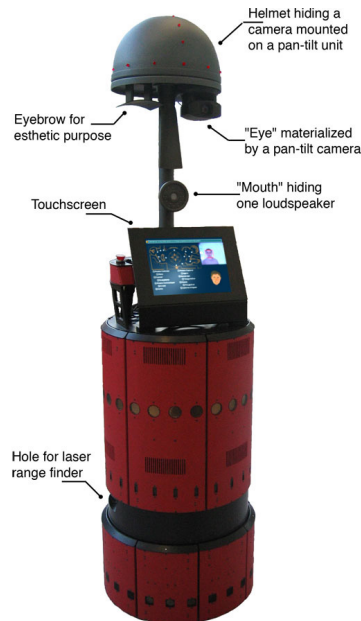


Figure 20: Rackham

The person presence and willingness to interact is then checked, by detecting whether he is gazing at the robot. If visual contact can be established, the robot comes nearer to the human. At any moment, the robot should be able to detect presence and commitment of the person and react accordingly. If the human shows no interest or has gone, the robot goes away.

B.3.4 Software Architecture

B.3.5 Implemented Services

The three following subsections present the main services that have been developed and implemented on the robot for the demonstration.

The first one covers two services: *ICU* (for *I see You*) enables the detection and tracking of people through a pan-tilt camera and *HumPos* locates more precisely people using a laser sensor.

The second one *NHP* (*Navigation in Human Presence*) allows to navigate in human presence: this path planner integrates "social rules" to produce robot paths that are "acceptable" by the humans in its vicinity.

The last sub-section will shortly describe the main supporting services: the services that are mandatory to perform the task but that are not addressed directly by the Cogniron Functions: *Localization*, *Navigation*, ...

1. Human Detection and Tracking Services

Purposes: These services are related to human detection and tracking using visual primitives. Our visual tracking strategy consists in fusing visual cues into particle filtering schemes. This fusion can be performed in the definition of the particles likelihoods. In addition, outputs from detection primitives can be involved in the importance functions of the particle filters.

So, the following detection services have been developed and integrated on the robot:

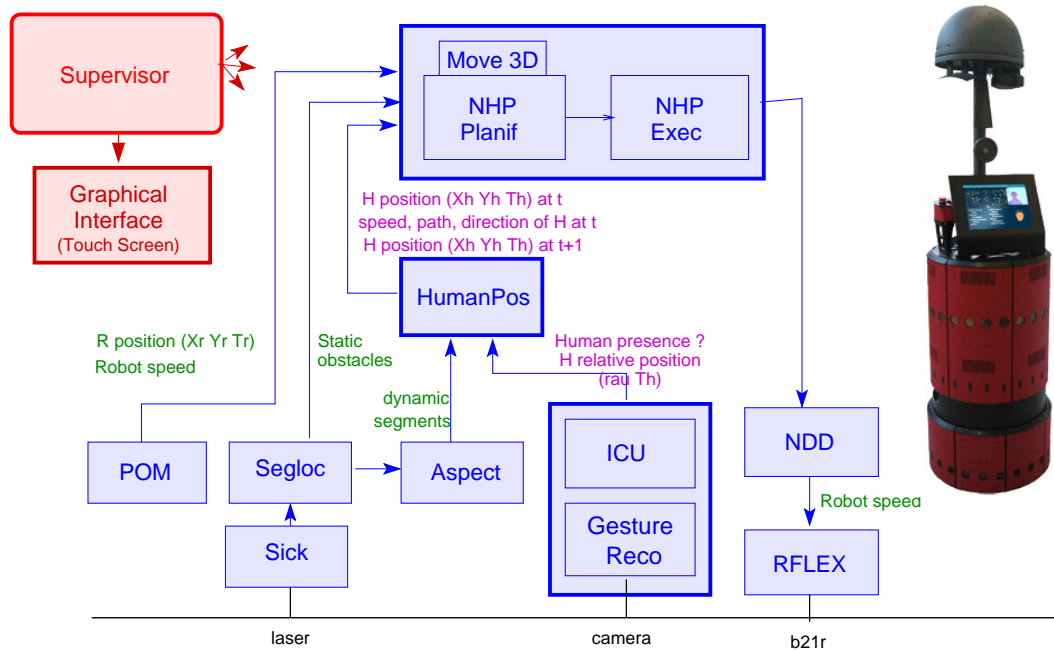


Figure 21: Software architecture with all the services involved

- a *face detector* computes the number of detected faces together with their positions and scales in a B/W or RGB image, using Haar-like features;
- a *face detector dedicated to tracking* computes a Gaussian mixture related to detected faces in an B/W or RGB image;
- a *motion detector dedicated to tracking* computes a Gaussian mixture related to the motion activity in a B/W or RGB image;
- a *skin-color blobs detector dedicated to tracking* computes a Gaussian mixture related to the detected color blobs in an RGB image.

Three visual tracking modalities were defined for H/R interaction, namely,

- a *head tracking modality* for interaction at short-range H/R distance (Figure 22 (a));
- a *human tracking modality* for interaction at medium-range H/R distance (Figure 22 (b));
- a *monitoring modality* for long-range H/R interaction (Figure 22 (c)).

All these modalities were integrated on the robot. Two kinds of videos¹⁹ are available, one showing an example of integration and the others demonstrating various particle filtering strategies.

Relationships with RA and CF:

This work is part of RA2 *Detection and Understanding of Human Activity*. The implemented 2D visual trackers were developed in the WP2.1 *Detection and Perception of body parts based on sensor features*, and are involved in the Cogniron function CF-TBP (*Tracking of Body Parts*.) The details are included into the RA2 deliverable.

¹⁹Videos are on the CD submitted with the deliverables 2005 (section RA7).

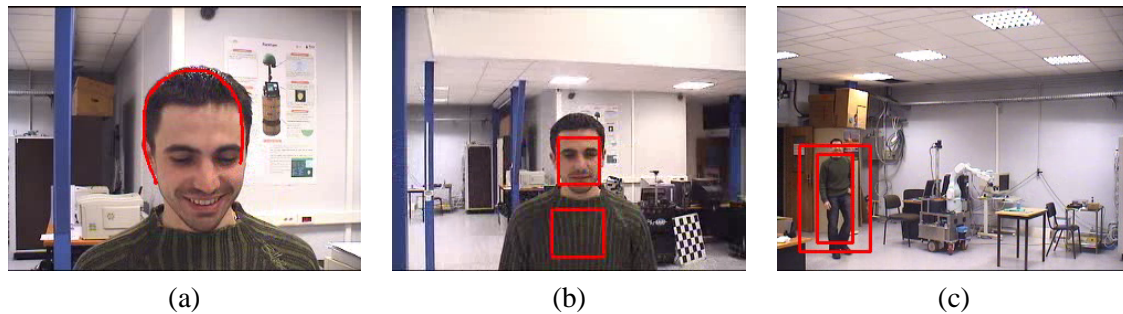


Figure 22: Modalities provided by the ICU module

Integration:

The ICU module has been implemented in C/C++. It provides some requests to control the embedded camera, and to manipulate the three previously described modalities. It can handle internally the switch between these modalities, based on the *face detector* results (frequency and size of the detections.) ICU can also approximate the location of a person at short or mid-range distance to the robot. All the tracking outputs from the various modalities are available to the other modules and more specifically for the supervisor.

Results:

A preliminary evaluation enabled to select the most meaningful visual cues associations in terms of discriminative power, robustness to artifact's (clutter or illumination changes) and time consumption, be these cues involved in the particles likelihoods or in the filters' importance functions. As a result, dedicated visual cues were selected for each modality, see below.

Several particle filtering strategies were evaluated in order to check which ones best fulfill the requirements of the considered H/R interaction modalities. The evaluated strategies are CONDENSATION, ICONDENSATION, the Auxiliary Particle Filter and the Rao-Blackwellized Subspace History-Sampling Sampling Importance Resampling (RBSSHSSIR) algorithm²⁰. For the sake of comparisons, importance functions rely on dynamics or measurements alone, or combine both. Further, each modality has been evaluated on a database of sequences acquired from the robot in a wide range of typical conditions: cluttered environments, appearance changes or sporadic disappearance of the targeted person, jumps in her dynamics... For each sequence, the mean estimation error with respect to "ground truth", together with the mean failure ratio (% of target loss), were computed from several filter runs. Some associated figure plots, as well as some tracking scenarios, can be found on the LAAS website²¹.

Eventually, the *head tracking modality* combines motion and shape cues into a CONDENSATION algorithm, the *human tracking modality* merges shape and color information into a Rao-Blackwellized Subspace History-Sampling Sampling Importance Resampling (RBSSHSSIR) algorithm, while the *monitoring modality* relies on color and motion distributions into an RBSSHSSIR algorithm.

Next step:

²⁰The CONDENSATION and ICONDENSATION algorithms were formerly developed by Blake and Isard in 1996 and 1998 [36, 37] respectively, the Auxiliary Particle Filter was defined by Pitt and Shephard in 1998 [38], and the RBSSHSSIR algorithm was developed by Torma and Szepesvári in 2003 [39].

²¹<http://www.laas.fr/~lbrethes/KE2.2k5>

Future theoretical developments in WP2.2 *Human body model: integration and fusion* will concern 3D human posture estimation based on an appearance approach. These will be integrated in the last phase of the project.

In addition, some marginal considerations concerning gesture recognition (Cogniron function CF-GR *Gesture Recognition*), under development in WP2.3 *Context-based interpretation and classification of activities*, will be integrated in a future scenario in order to interact with the robot through communicative gestures.

2. Navigation in Human Presence Service

Purposes:

This service contains an implementation of the Human Aware Motion Planner (HAMP) developed in the framework of RA3 (WP3.3). This planner computes “acceptable” and human-friendly paths by reasoning on humans state, their vision field and their accessibility. Three different criteria have been extracted from user studies for these human-robot interaction and serve as the core of HAMP. Each of these three criteria is represented by a 2D grid attached to each human containing various costs depending highly on humans’ states in the environment.

- **Safety Criterion:** The robot must avoid to collide with the human in the environment, and if possible it must prefer to not to pass too close to him. This criterion is represented by a human centered bell shaped grid.
- **Visibility Criterion:** The robot, if possible, must prefer to stay in the humans field of view. This criterion allows to take into account the humans vision field and allows us to produce more “acceptable” robot motions. The visibility grid is constructed according to costs reflecting the effort required by the human to get the robot in his field of view.
- **Hidden Zones Criterion:** The robot, if possible, must avoid to burst near the human to avoid to cause fear and surprise. This is done by putting costs to the zones hidden by the obstacles. The costs in the hidden zone grid are inversely proportional to the distance between the human and the robot.

Not only the shape of each grids differs one from another but also each human in the environment can have different parameters, so different shapes, for the same type of grid. These grids are combined to form a final grid in which we find a minimum-cost trajectory for the robot by using A* search algorithm. The synthesized path is not only collision free and feasible but also human friendly with satisfying all three criteria. One must note that only the final grid, on which we conduct A* search, is constructed explicitly. The cost of a cell in the other grids is calculated in case of a request from A*. This approach allows us to fasten the planning process and limit memory space usage.

In the next sections we will focus on the implementation aspects, but the detailed inner working of the system can be found in the deliverable D.3.6.

Relationships with RA and CF:

See Deliverable D.3.6 - Models and algorithms for motion in presence and in the vicinity of humans.

Integration:

Although all the aspects of Human Aware Motion Planner are not implemented, most of the major functionalities are currently working on the robot. As can be seen in the service's architecture schema (figure 23), Move3D and grids build the core of the motion planning module. This module works with a static internal 3D map along with each humans model, his grid construction parameters and the robot model. The humans' positions are updated by the humPos service and the robot's current position is updated by Pom service. The motion planner module then calculates a path and sends it to the path execution service.

As the planner does not take into account the dynamic obstacles and does not update his map, we compensated this gap by using a complementary service, namely NDD, that executes and modifies dynamically the path produced by NHP. But a very next step in implementation will be to allow the planner update his environment and take into account the dynamic obstacles.

The implementation of this service is done in C, like Move3D and grids algorithms and in the form of a "module" in LAAS architecture.

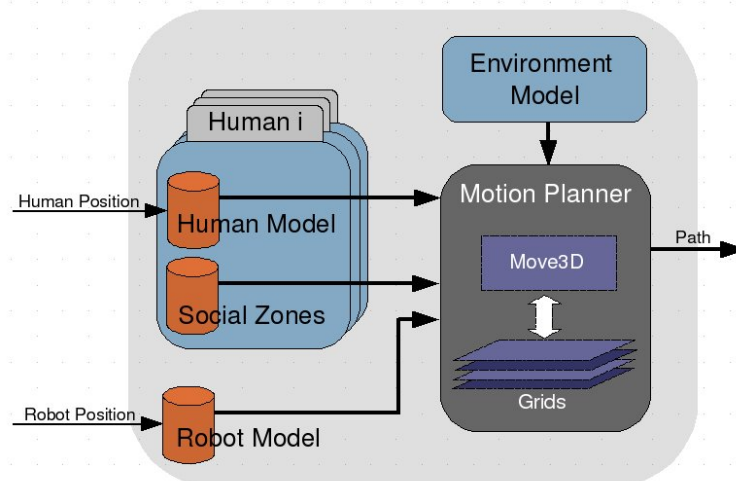


Figure 23: NHP Service implementation architecture

Results:

The results of current system are no mature enough to cite in this deliverable, but the simulation results of the experiment environment are encouraging and planning process times are short enough to plan/replan online (Deliverable D.3.6). In the figure 24, we can see two different situations with four humans in it. The planner produces a path that take into account the safety and the comfort of each individuals in the environment. Videos²² showing simulation results are available.

Next step:

Next steps of the implementation will mainly focus on the dynamic changes in the environment and re-planning as a result of these changes, along with the algorithmic improvements. The feedbacks of the service to the supervisor need also to be improved and enriched. According to the algorithmic improvements, the robot's speed will be handled by the NHP and a switch

²²Videos are on the CD submitted with the deliverables 2005 (section RA7).

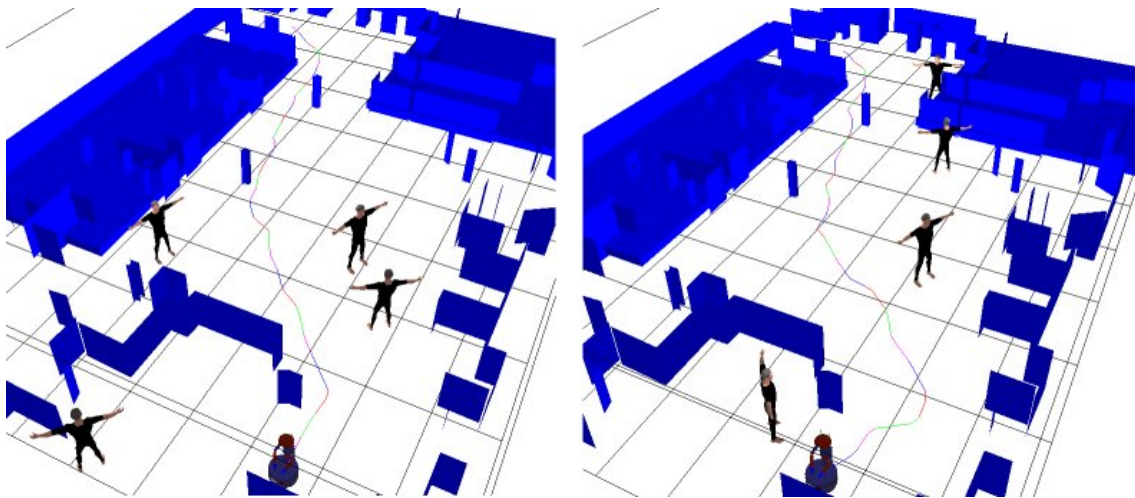


Figure 24: Two scenarios containing four humans in the environment. The path found by the planner is computed by taking into account each humans vision field as well as their safety and the zones hidden by the obstacles.

mechanism will be implemented between NHP service and a pro active planner (see Deliverable D.3.6) that explicitly into account the dynamics of the environment.

3. Supporting Services

Localization with sick, segloc and pom:

To localize itself within its environment and detect obstacles, the robot uses a SICK laser, (*sick* module). From the laser echoes together with segments deduced from aligned echoes the module *segloc* build the map and localized the robot within this map thanks to a classical SLAM procedure.

The *pom* module allows to merge the robot positions (from *segloc*, the odometry and the gyroscope), taking into account their uncertainties, and to export one and only one reference position to the other system components.

Obstacle Avoidance with aspect and NDD:

Rackham's laser can only look forward (over 180 degrees) in an horizontal plan. To partially overcome these limitations, the laser data are integrated in a local map updated at 40ms by the *aspect* module and filtered using knowledge about the global map.

The module *ndd* integrates a local avoidance procedure, within this local map, based on an algebraic instance of Nearness Diagrams [40].

Human Positioning with humPos:

As the robot's field of view is narrow and the range of human detection by ICU Service is limited, this service is specifically created to produce an input on humans' positions in the environment for the NHP Service by using laser information. The human detection by *humPos* is made in three stages:

- Leg detection in *aspect*: Segments, that do not exist in the environment, obtained from

aspect service are used for the recognition of possible legs. All segments that pass through this filter are tagged as possible humans with an associated confidence value.

- Filtering by using `sick`: Groups of points obtained directly from laser have a number of properties and shapes if the points indeed belong to a leg. This stage of the process applies this filter directly to the previous stage's possible humans' raw laser data. At the output of this stage, the shapes tagged to be humans have a higher probability to be so.
- Visual Detection: If the potential humans are also detected by the vision based service ICU, their associated confidence level humans increases dramatically.

Another property that has effects on `humPos` is the motion of humans in the environment. If a shape is not moving for a predefined time, it is thrown away and tagged to be not human even though it had a high probability. On the contrary a shape that moves in the laser map, has a higher probability to be a human.

Detected humans along with their positions, orientations and their probability are transmitted to the NHP Service. This service is implemented in C as a module of the LAAS architecture.

B.3.6 Supervision for Human-Robot Interaction

Purposes: A supervision system based on joint intention theory that will allow to take into account the human-robot interaction at the planning and execution level.

For this, each human entering in the field of the robot is taken into account as an agent with whom the robot can collaborate or at least interact. To be homogeneous, the robot itself is also considered as an agent. Each agent is characterized by its abilities (tasks types that it can perform), its potential commitment to a task at hand and once involved, observers help to follow each agent involvement and state towards the task. This information is then used by the robot to achieve its task and to adapt it to human action, reaction or lack of reaction.

Two other considerations have been taken into account:

- multi-modality: in HRI, it's often the case that a given information comes through various ways (e.g. a human can give the same order by gesture, by speech, by the help of a touchscreen or by a combination of these modalities).
- re-usability: new functionalities are available along the project, we argue that our supervision system is built in a modular way to be able to integrate incrementally new robot abilities.

Relationships with RA and CF:

This is linked to CF-RET *Reasoning about Tasks*. See also Deliverable 6.4.b: Models and Algorithms for Cognitive Robots Supervision and Deliverable D6.4.c Human-Robot Collaborative Problem Solving.

Integration:

We have developed and implemented an agent-based supervision system that deals with tasks in terms of individual tasks (only the robot is involved), joint tasks (the robot and another agent are involved) and activities that corresponds to low level functionalities that are not further decomposed.

Each task is defined by a plan and dedicated monitors. A plan corresponds to a succession of sub-tasks and/or activities. Monitors serve to state whether a task is unachieved, achieved, impossible, irrelevant or stopped.

Consequently, the system that can be controlled at different levels at the same time. If something is detected at a given level, the system is able to take it into account at that level by applying adapted solutions and propagating, when necessary, events towards the higher or the lower levels.

In the current implementation, the supervisor is written in Open-PRS. The task plans are hand-coded (a set of pre-defined task library) and only the robot is able to propose a task. However, the supervisor is built, taking into account future extensions involving on-line task planning.

Concerning joint tasks, their associated plans include several steps that allow to respect the joint intention scheme:

- PRETASK: concerning the establishment of the joint task between a person and the robot,
- INTASK: concerning the progress of the defined task,
- POSTTASK: concerning the end of the task. Joint intention theory requires that all agents must be informed of the end of the task,
- CHECKTASK: concerning the monitoring of the agents commitment to the task.

Each of these *TASK can be defined as an individual task, a joint task or an activity.

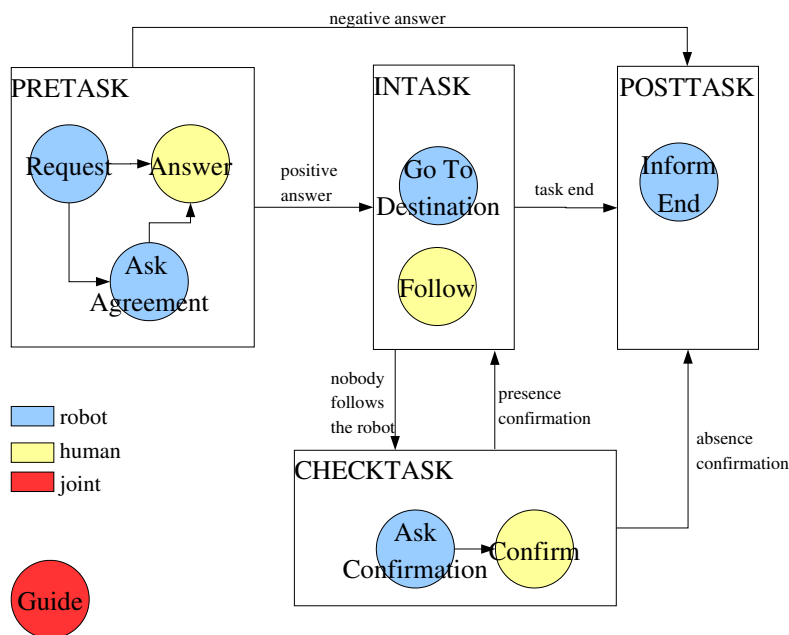


Figure 25: A simplified view of the joint task: Guide

Results:

This supervision scheme has been tested on Rackham during its last stays at the Space City Museum in Toulouse. The root plan of the robot was: *Initialization, Looking for a visitor and Guide visitor*. Initialization was only done once, after the robot switched between the last two tasks.

Figure 25, gives an overview of the joint task *Guide visitor*.

During four days, Rackham ran in average one hour per day (the only time limitation was the presence of a Space City animator). 1/4 of the time the robot are *Looking for visitor* and 3/4 of the time,

Rackham *Guide visitor*. A commented video²³ of this demonstration is available.

Moreover, we have used the same system to integrate Human-Detection and Tracking Services that have been developed for Cogniron. In that case, the plan was: *Initialization, Wait For People, Want to interact ?, Do Service*.

Wait For People and *Want to interact ?* are two complementary tasks. *Wait For People* detects by the help of the Motion Monitoring if someone enters in the field of the robot. If it happens, *Want to interact ?* checks if the person is interested by the robot i.e. approaches the robot (Body Tracking, Face Tracking and finally Face Detection) and/or touches the touch-screen etc. If nothing happens for a while in this second task, the robot returns to *Wait For People*. Otherwise, *Do Service* is launched. *Do Service* is a version of *Guide Visitors* that uses all the services provided by Human-Detection and Tracking Services whereas *Do Guide* just uses Face Detection. A commented video²⁴ of this demonstration is available.

Comparing to the previous system, it is easier to add new capabilities to the robot and to program new tasks involving interaction. In addition, monitoring is facilitated by the help of layered levels that are all reactive.

Next step:

In the near future, the system will be improved with the integration of an on-line planning ability that explicitly considers Human-Robot Interaction and Collaborative Problem Solving. That will allow us to give not only short-term but also long-term view/plan to the robot. Concerning the supervisor itself, we envisage to extend it in order be able to take into account more than two agents (the robot + a human) by task and to manage several tasks simultaneously at the robot level.

B.3.7 Results and Conclusions

Work done in 2005:

- Definition of a set of Cogniron services (and their interaction) related to motion in the vicinity of humans
- Development and integration of a “Human Aware Motion Planner”
- Development and integration of a first version of robot supervisor specially dedicated to human-robot interactive task achievement
- Development and integration of a set of vision-based human detection and tracking services
- Development and integration of several supporting services

The overall experiment has not yet been demonstrated. However, all essential ingredients are now implemented.

B.4 Experiment KE2-3: Curiosity and Learning

B.4.1 Objectives and Context

In this experiment we want to investigate knowledge acquisition in terms of sensori-motor representations and with little *a priori* knowledge or representational choices to preserve genericity and open

²³The video is on the CD submitted with the deliverables 2005 (section RA7).

²⁴A video is on the CD submitted with the deliverables 2005 (section RA7).

endedness. The purpose of the experimented approach is to enable to build representations associating perception with action through an unsupervised learning process during which the robot interacts with its environment. The curiosity behavior which drives the robot to explore newly perceived objects is related to increasing a value function. The underlying architecture we are experimenting here is a neural one as explained below. This architecture enable to implement in a unique manner the perceptual and motor representations, which are linked together. It also provides for distributed and hierarchical representations, starting from simple image processing, which we believe are a good basis for achieving open-endedness without being constrained by specific primitives. The Cogniron Functions implemented in this experiment is **OR** (Object recognition), embedded in the learning process. One important question that arises then is the possibility of integrating such an architecture with the symbolic-oriented ones. This issue will have to be investigated further in the next period. One approach would be to introduce a neuronal-symbolic interface that translates the neuronal representations into predefined data structures by means of a supervised categorization process in which the human tutor names the categories.

B.4.2 Robot Platform



Figure 26: Athos is a two-wheeled robot equipped with a pan and tilt stereo head.

Athos is a SuperScout 2 robot built by Nomadic Technologies. It is a 35 cm wide and tall cylinder equipped with two Sony DFW-VL500 digital cameras mounted on a Directed Perception pan-tilt unit. Its has also 24 sonar range sensors which are not used during our experiments. The robot motion controller and the Pan-Tilt unit are both connected to a Apple Powerbook G4 867MHz for video acquisition, pan-tilt and motion control via USB ports.

B.4.3 Scenario

In this simple set-up, the robot observes the floor and upon detecting a new object, it moves the stereo head, and then its full body to learn the object. The representations associate the views with the actions which are not initially known. A multi-level knowledge is acquired from camera acquisitions and proprioceptive sensor data coming from pan-tilt unit and motor controller. The curiosity behavior is generated during the exploration of these perceptive data.

B.4.4 Software Architecture

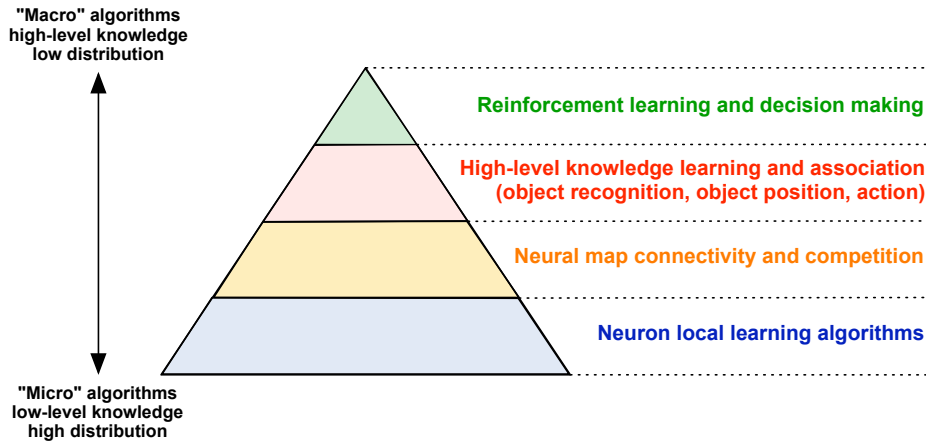


Figure 27: Global architecture for knowledge and skill acquisition composed by a set of algorithms hierarchically structured.

In this section we will describe a global neural architecture for knowledge and skill acquisition. As illustrated in Figure 27, this architecture can be viewed as a set of algorithms hierarchically structured. At a bottom level, the algorithms simulating neuron activations are local and highly distributed (several thousands neurons, million connections), and compute very low-level knowledge as pixel contrasts (see section 1). In a second level, neural maps which are sets of hundreds neurons are connected in a hierarchical way and are in competition for higher level knowledge extraction as pattern recognition (see section 2). In a third level, three distinct sets of neural maps encoding object representations, object positions and robot's actions are associated to generate predictive models of actions (see section 3). Finally, a global reinforcement learning algorithm provides a mechanism for choosing the most relevant action in order to increase a global system score we call the global effect (see section 4).

1. Neurons and Local Algorithms

Fundamentals:

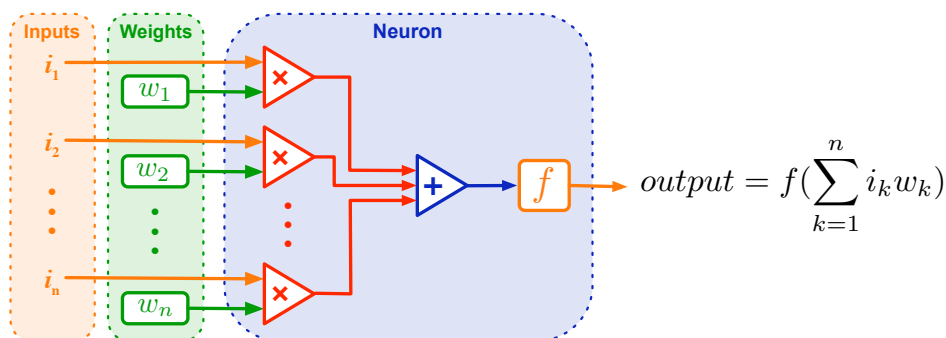


Figure 28: Simple neuron with its sets of inputs i_k and weights w_k .

A simple neuron is shown in Figure 28. It computes the sum of each product of (i_k, w_k) pair. Inputs and weights are considered between -1 and 1. An increasing monotonic function f , called

transfer function is then applied and the result, also considered between -1 and 1 is available. A threshold can be also applied that permits a binary behavior for strict decision making. The function of a neuron can be viewed as the partition of a n-dimensional space where n is the number of inputs. The weights define a n-1 dimensional hyperplane through the origin, and depending on the values of the inputs, the point is below or above the plane.

Integrate and Fire Model:

When considering neuron activations across time the previous simple neuron model can be refined. Next equations describe our own implementation for a discrete time model and have been simplified for better understanding. Previous formula shown in Figure 28 now become:

$$output(t) = f(t)(P(t)) \quad (1)$$

where $P(t)$ is called the potential at time t and is given by:

$$P(t) = \alpha_P P(t-1) + (1 - \alpha_P) \sum_{k=1}^n i_k(t) w_k(t) \quad (2)$$

α_P is a potential leak and offers a way to represent the memory trace of previous activations. At each time step, the updating of potential $P(t)$ is called the integration phase. Then the neuron "fires" when its potential crosses a given threshold $T(t)$. In our model both threshold $T(t)$ and transfer function $f(t)$ are tuned based on statistical observation of neuron activations. This mechanism won't be explained here.

Hebbian learning rule:

We use in our model a Hebbian-like learning rule in which neurons tend to learn patterns responsible for their activation. In other words, a learning process occurs when one neuron fires. Moreover, we compute a stochastic standard deviation σ_k for each weight w_k which is computed and used as coefficient in the learning calculation as follows:

$$\mu_k(t) = (1 - \alpha_W) \mu_k(t-1) + \alpha_W | \gamma_k(t-1) | \quad (3)$$

$$\sigma_k(t)^2 = (1 - \alpha_W) \sigma_k(t-1) + \alpha_W [\mu_k(t) - i_k(t)]^2 \quad (4)$$

$$w_k(t) = (1 - \alpha_W) w_k(t-1) + \alpha_W [1 - 2\sigma_k(t)] i_k(t) \quad (5)$$

Where μ_k is a stochastic mean and α_W is the learning rate. Hence we ensure that weights corresponding to noisy inputs (with high standard deviation) will tend to 0 and then won't take part in the neuron's activations.

2. Maps, connectivity and competition

Maps: As illustrated in Figure 29, a map is a collection of neurons sharing the same weights. Each neuron is composed as a three stage calculation pipeline and receives signals incoming from neurons situated on previous layer for integration and lateral signals from maps situated on the same layer for competition. Learning weights are organized in kernels, and there are as many weight kernels as afferent maps (see Figure 31). All the neurons of a map are sharing the same set of weights, thus they can detect and learn a pattern which is at different positions in the input maps. As shown in Figure 29, the coordinates of a neuron in a given map and its receptive field in the afferent map are the same. Therefore the position of an active neuron corresponds to the position of the detected pattern, which provides for a shift-invariance property.

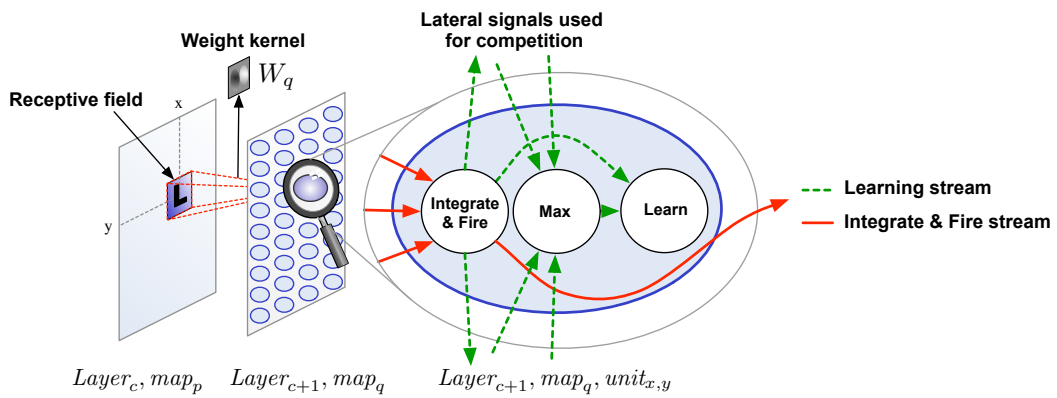


Figure 29: A map is a collection of neurons sharing the same weights. A neuron is composed as a three layer pipeline.

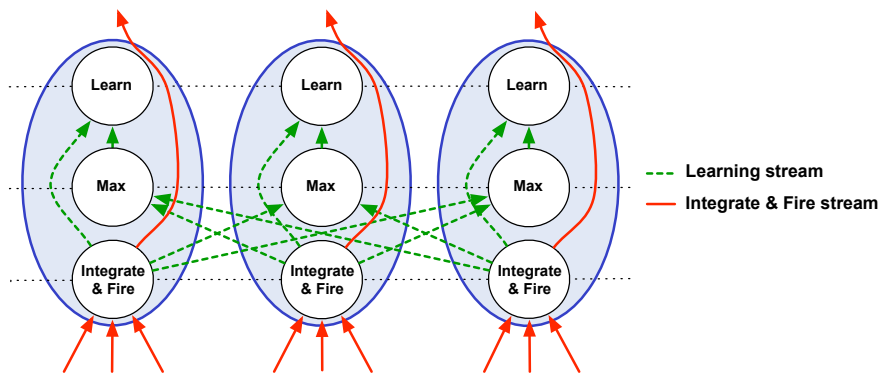


Figure 30: Three neurons connected with lateral connections used for the competition process between maps.

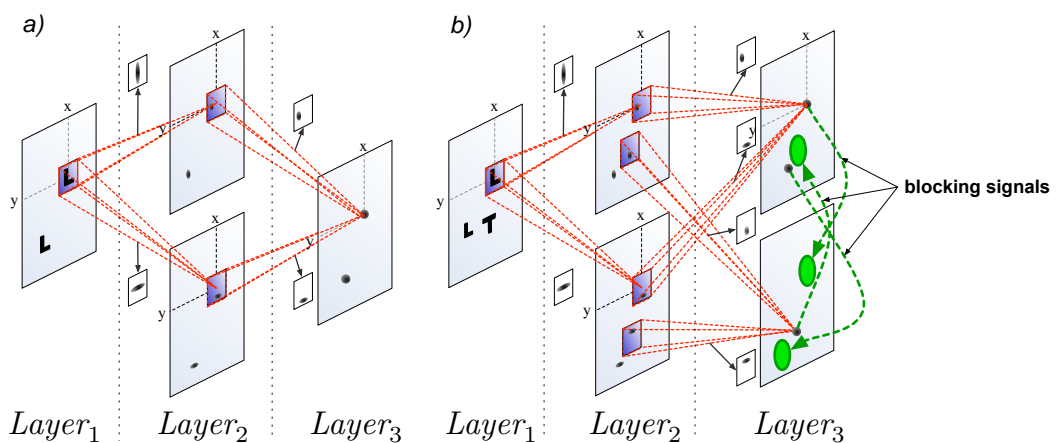


Figure 31: a) Hierarchical maps connectivity for pattern extraction. b) Blocking signals during maps competition.

Connectivity: We adopted the layered hierarchical architecture shown in Figure 31 for several reasons, first we need this architecture for categorization. As our neurons proceed to linear separation in their input set, complex features cannot be extracted with a single layer. Another reason is reusability. A huge number of extracted features are encountered in different shapes. Oriented segments, arcs of circles or color blobs are building blocks of more complex images. As extracted information is shared in the network, we prevent our system from computation redundancy.

Competition: For competition, every neuron is connected to other neurons situated at the same coordinates in different maps of the same layer. Figure 30 focuses on the competition process involving these neurons. The 'Max' stage, as its name indicates, computes the maximum incoming output value and sends it to its own learning stage. The learning process is then able to compare the local neuron activation to distant activations. By allowing learning only to the best fitted neuron, we ensure that no map could learn one pattern if another is already specialized to detect it. We illustrate this inter-maps competition in Figure 31.b with the green discs which represent the blocking signal.

3. High-level knowledge acquisition

The third algorithmic level (red in Figure 27) is based on the neural maps capability of extracting autonomously two dimensional hierarchical representations in their input space. We will here present the three classes of maps we use to learn object appearance, object position and robot's actions. White arrows in Figure 33 illustrate the autonomous map creation process which permits to increase the size of the neural network when new knowledge needs to be learnt.

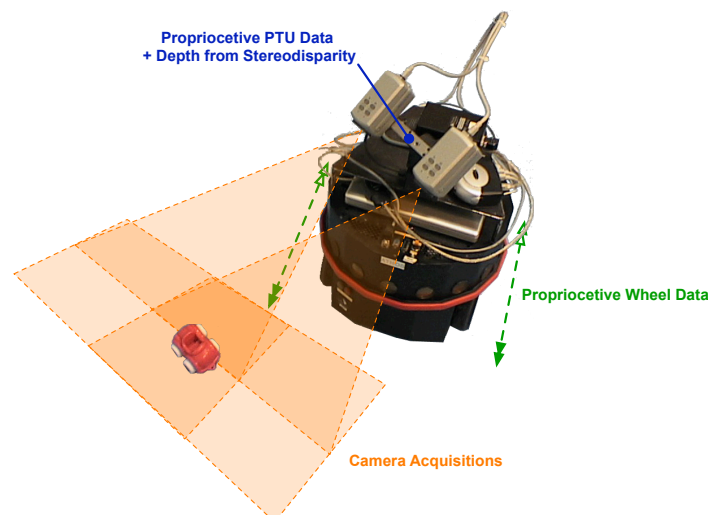


Figure 32: Three classes of sensor data.

View-based object representations: We use stereo camera acquisitions to learn object appearance. As shown in Figure 33, each object is represented as a collection of view-dependent detectors. These detectors are trained maps connected to orientation detectors in a way illustrated in Figure 31.b. Two kind of autonomous map creations exist here: "A" for new view detectors and "B" for new object detectors.

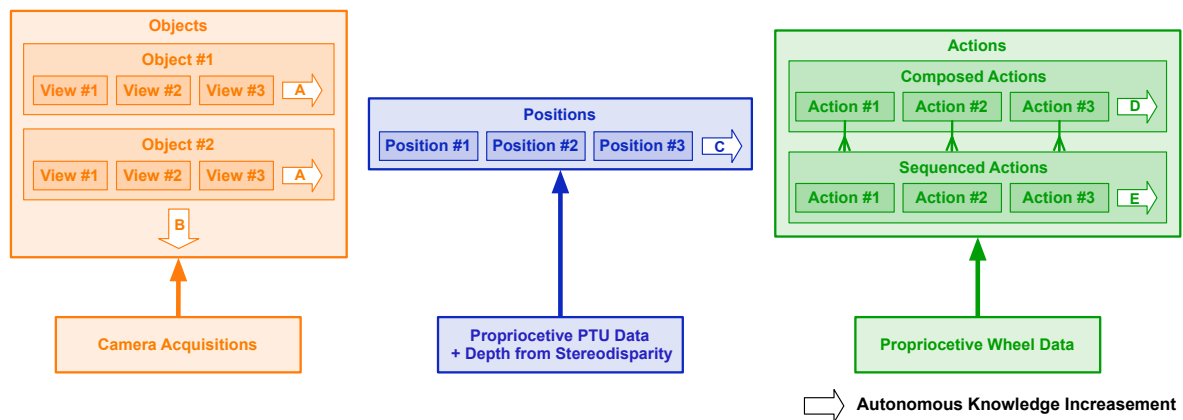


Figure 33: From sensor data to high-level representations. Object representations, object positions and robot’s actions are learnt separately in three distinct classes.

Head-centered object positions: While detecting an object, a tracking module is activated to center the object in front of the two cameras. Then, some dedicated maps receive as input three proprioceptive maps whose respective activation patterns reflect pan position, tilt position and stereo-vision disparity. After a short learning period, each map is activating for one specific position of object. Map creation "C" is necessary to increase the number of known positions.

Composed and sequenced actions: In a similar way, a third class of maps is trained to recognize and produce patterns of action. As no knowledge about actions is a priori known, the robot initially needs to produce random actions leading to random commands in velocity to its wheels. These commands are recorded and used as proprioceptive data and inputs for specialized maps. Thus, each map learns one action composed of a couple of velocity values. Knowledge which relates to composed actions is increased by "D". As several composed actions are learnt, a second class of maps learn to recognize and produce sequences of actions. Composed actions are then used as input in special maps organized in timelines. Finally, sequenced actions are learnt as temporal patterns of composed actions and created by "E".

Associations and predictive models:

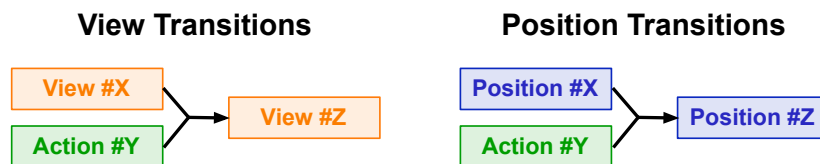


Figure 34: Two kinds of associative scheme are defined to predict actions consequences on perceptions.

Using the three knowledge classes exposed before, we propose the associative scheme illustrated in Figure 34. The first association aims to create a relationship between the representations of object’s views mentioned before and can be considered as the construction of an aspect graph. As the robot moves, it is able to both represent its actions and the views of the object (respectively orange and green color in Figure 33). Then an association can be learnt involving the representation of the action the robot just achieved, the representation of the view

recognized before the action started and the representation of next view activated by the action. This kind of association involving robot's actions and object's views constitutes a sensori-motor representation of the object and can be used as a predictive model of actions.

The second association shown in Figure 34 and called Position Association is built in a same way with actions and position representations. We propose to separate object's appearance and object position in order to reuse this second type of transition for any object and then avoid combinatorial explosion of the number of representation.

4. Reinforcement learning and decision making with the *global effect* mechanism

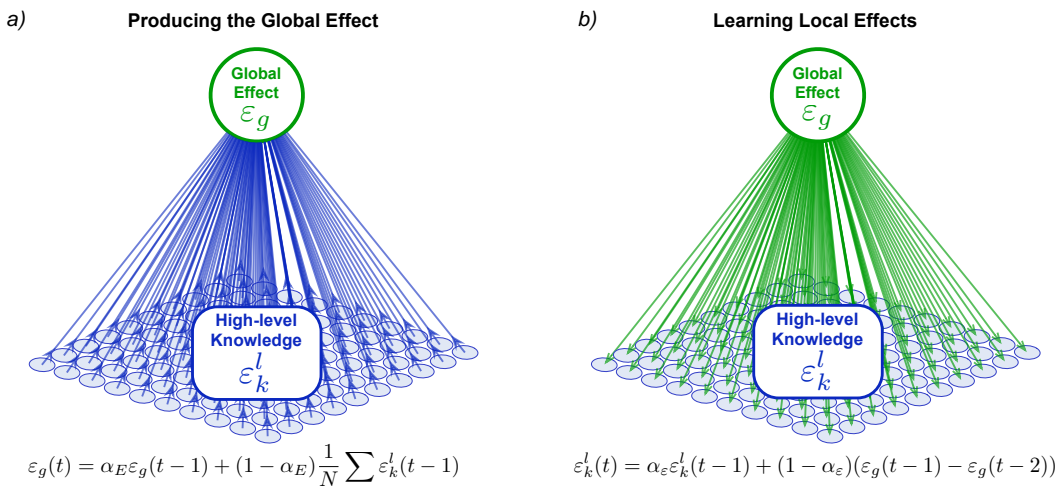


Figure 35: a) Each high-level neuron activated at time t contributes to the global effect update. b) During the learning phase the active neurons learn the global effect variation.

We define the system's purpose as the satisfaction of a given criteria. The ability of learning new goals implies that new related criteria have to be defined by the system itself. We need the system to find its own criteria. Moreover, satisfying a criteria cannot be done without evaluating the new context after an action is done. So we need to rate perceptive contexts during system experience in the environment (scoring). As shown in Figure 35.a, each representation component (or neuron), is able to emit an effect ε_k^l when it has been activated. Therefore, values are attached to the neurons of each representation. They become the components of a function the system wants to maximize: the global effect ε_g . The goal of the system during its operation is to activate the representation that maximize ε_g . By extension, these representations are called the goals. They correspond to situations in the real world that the system wants to reach. This global value ε_g can be positive or negative, depending on the representations. Initially the system has only a small set of representation goals with hard coded effects associated with it, some with a positive effect (the basic goals it wants to reach), and some with negative effect (the contexts the system has to avoid). These representations give the first structure of system behavior. The other representations are neutral. New goals, or representations to avoid, are obtained by reinforcement rules and inherit of all the representation components that appear simultaneously with the global effect ε_g (see Figure 35.b). Activating one of these new components becomes a potential goal, or a representation to avoid for the global system.

The system's global effect also aims to select relevant representations that need to be learn. We don't need to learn every possible object positions or robot actions if they don't provide a way

to increase the global system criteria. Finally, as associations shown in Figure 34 also take part in the effect learning process, the system is able to select at each step the one that produces the maximum increase in global effect.

B.4.5 Implementation

Low-level algorithms simulating neuron activations have been implemented in a simulator called NeuSter which can be used as a line command tool. This software enables real-time computation of neuron activation, image acquisitions from stereo cameras, motor control and in-line neural maps creation. Moreover, neural networks are simulated and distributed on multiple POSIX machines connected in a local network to increase performance. Another software called Assemblee has been developed to monitor neuron activations, create and save network in a user-friendly visual interface, tune parameters and manage clusters of machines. Higher level algorithms have been tested using these tools and are presented here. Although the object, position and action representations are here systematically learnt to demonstrate unsupervised learning capabilities of our algorithms, they need to be driven by the global effect mechanism as explained in section 4.

5. View-based Representation of Objects

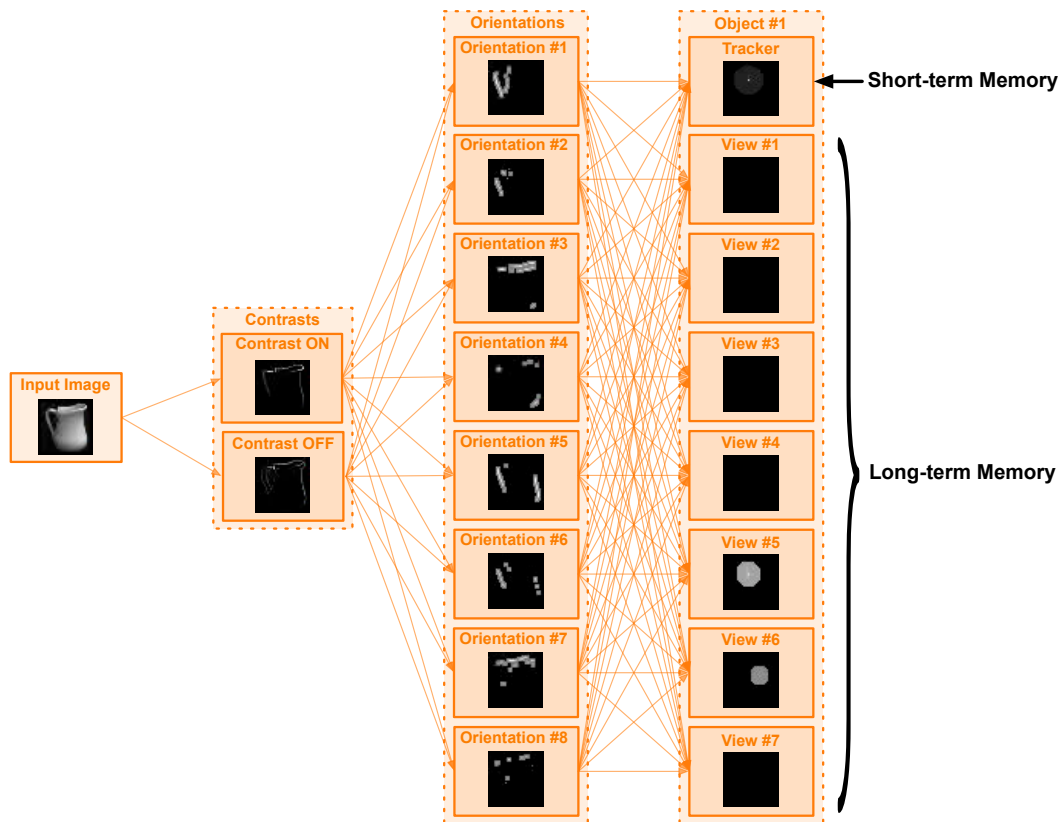


Figure 36: The input image is decomposed through contrasts and orientation detectors. Then, long-term memory maps learn to recognize one specific object orientation while a short-term memory map called tracker follows the position of the object.

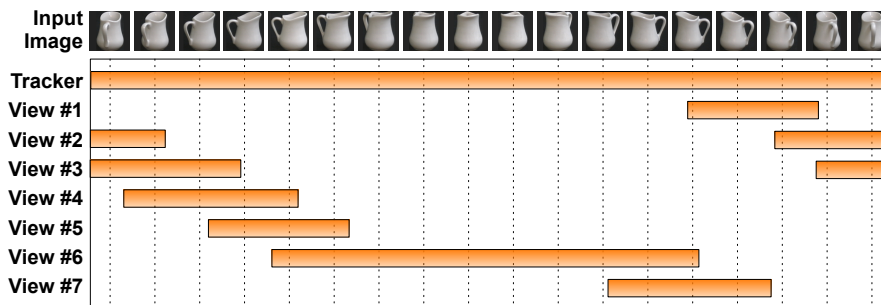


Figure 37: Activity diagram showing views detection and object tracking during rotation.

Purpose: As presented in 3, we want to extract high-level representations of object appearance. Each object is represented as a collection of views with neural-maps connected to a set of maps extracting local orientation in the input image.

Results: Figures 36 and 37 illustrate our results. We used inputs from COIL database where images of objects from various orientations were acquired. The trained neural network was able to recognize an object from various view points. We propose used two kinds of maps with different learning rates for both tracking (short-term memory) and recognition (long-term memory). These results have been published at the 19th International Joint Conference on Artificial Intelligence [19] (IJCAI-05) and contribute to WP5.2 *Models of Objects*.

6. Head-centered Representation of Object Positions

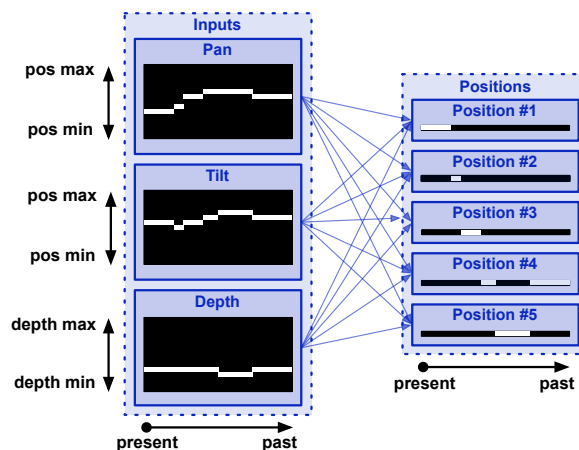


Figure 38: A neural network can be trained to learn position patterns in maps structured in timelines.

Purpose: This algorithmic module concerns the autonomous learning of objects position. The inputs of this neural network are acquired from pan position, tilt position and stereo-disparity when an object is detected and centered in front of the two cameras.

Results: Thanks to our competition process between neural maps (see section 3), the network shown in Figure 38 is able to learn position patterns while observing proprioceptive input data. We here used maps behaving as timelines to create the inputs.

7. Composed and Sequenced Actions

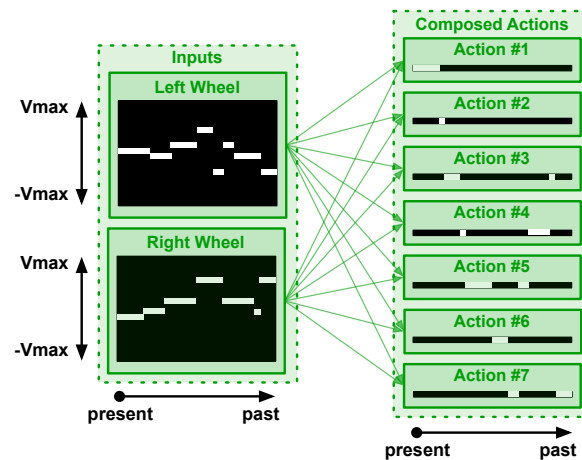


Figure 39: Results of composed action learning.

Purpose: Action representations are the third class of high-level knowledge (see section 3). This knowledge must be acquired from proprioceptive data as it is not provided to the system initially. In a second phase, the decision system will have to use these representations to produce underlying actions in order to increase the global effect.

Results: Figure 39 shows results of composed actions learning where the inputs are proprioceptive data coming from randomized commands sent to wheels. We show that our low-level learning mechanism and connection scheme are able to categorize composed actions.

Next steps: As we can categorize composed actions in timelines, we now plan to add another layer to the network shown in Figure 39 and learn sequences of actions.

8. Sensori-Motor Associations

Purpose: For decision making, the system needs to associate perceptive contexts and actions. When the global effect increases, new associative neural maps must be created to memorize the action to achieve in given context.

Results: The work was mainly done with position transition and proved the possibility of extracting relevant associations by using low learning rates and maps structured in timelines. With low learning rates and thanks to our statistical learning rules we were able to filter correct associations. Nevertheless this work is not achieved and needs to be combined with the reinforcement learning mechanism to be validated.

Next steps: We still need to validate the associative mechanism with view transitions and sequenced actions as soon as they will be available.

9. Reinforcement Learning and Decision Making

Purpose: The reinforcement learning mechanism is situated at the top level of our architecture and provides a way to select actions leading to contexts that increase the global system criteria. This algorithmic module is composed by the local effects learning phase and global effect production and by the action selection phase.

Results: The local effects learning phase and global effect production have been tested and are working properly. We successfully associated positive effects to one specific object detection

and were able to reactivate this positive effect when the robot detected the object again. Nevertheless we were not able to demonstrate the action selection mechanism as we first need finalize transitions learning.

Next steps: Demonstrate the action selection mechanism.

B.4.6 Results and Conclusion

Work done in 2005:

- Development of the neuron activity monitoring and network creation software: Assemblee.
- Validation of the object learning and recognition algorithms (IJCAI-05).
- Development of dedicated neural networks for positions and actions representations.
- Preliminary work on sensory-motor association.

Results on learning composed actions are encouraging and we now investigate the learning of sequenced actions. During year 2006, we will focus on the action selection and production level.

B.5 Conclusions

As already mentioned, 2005 has been essentially devoted to the development and integration of the basic ingredients on which KE2 experiment will be constructed.

A methodology for integration has been defined and several tools have developed and/pr deployed to allow the incremental development and integration of the different robot functions.

Partial integration has also been performed. The first Cogniron services have been encoded and implemented on the platforms.

Substantial ingredients are now available that will allow to build and demonstrate several abilities involved in KE2.

C Table of Cogniron Functions

Having identified the Cogniron functions we refined them in terms of Cogniron services. However, also we aimed at a first classification of the functions (see section C.1), first testing methods and many technological integration issues. The classification yields that actually most of the Cogniron functions are “perceptual” functions and many of them are related to learning. This has deep implications for the upcoming Cogniron architecture. We are currently discussing how the various perception and learning abilities can be “neatly” incorporated into existing three-layer architectures. As a side effect of this study we also found that the robot companions needs about 30 GHz processing power if all functions would be active at a time.

Section C.2 gives an overview of current (end of phase 2) CF implementation and testing status. The abbreviations in the “Category” column are:

I: Interaction

P: Perception

L: Learning

MG: Motion Generation

AD: Autonomous Decision making

C.1 Summary of Questionnaire Results

CF	KE	Partner(s)	State of development	Sensors(s)	Actuator(s)	Input		Output		Modes	Category				
						Contents	Source	Contents	Destination		I	P	L	M	A
DLG	1	UBI	First approach running, new version in development	Camera (on pan/tilt unit) Stereo microphones Laser range finder Stereo camera	Mobile platform Speakers Pan/tilt	Recognized words	Speech recognition	User instruction	Execution supervisor		X				
ROR	1	UBI	In development (basic module running, no interfaces working yet)	Camera (on pan/tilt unit) Stereo microphones Laser range finder Stereo camera	Mobile platform Speakers Pan/tilt	Objects Gestures Task	OR GR DLG	Resolved reference	DLG	Known Object Unknown object	X				
PTA	1	UBI	Running (new modalities can be added easily)	Camera (on pan/tilt unit) Stereo microphones Laser range finder Stereo camera	Mobile platform Speakers Pan/tilt	Operation mode	DLG	Information interaction partner (XML)	Platform (motion commands)	Sleeping Awake Alert Listening Interacting Following	X				
GR	3,2	UKA	Under development	Stereo Camera, TOF depth sensor	Pen-tilt unit for 3D Sensors	Sequence of geometrical human configurations	TBP	Type (label) of gesture and numerical parameters	DLG or other HRI components	Only one	X				
IA	1,2,3	UH	data on relationships between subjects personality and their perceptions of robots, data on the role of a robot companion in the home	UH is not providing software modules, so this does not apply.	N/A	N/A	N/A	Parameters and design guide lines	All Cfs which address interaction.		X				
LOC	1	KTH	An initial version is under testing	Sonar Laser range finder Color camera	Mobile platform	Direct from sensors graph/plan map in KTH format		Topological map (graph) + 2D planes in KTH format	NAV	Only one	X				
	1	UA	Running off-line	Omni-directional Camera	Mobile platform	Labels of locations	DLG	Estimated label	DLG, NAV	Only one	X				X

NAV		UA	Under development	Omni-directional Camera Odometry	Mobile platform	Labels of locations	DLG	Motion commands	Platform (motions commands)	Learning and operation	X	X	X	X
SOC	1	EPFL									X	X	X	X
	1,2,3	UH	data on social approach distances, approach directions and subject's comfort levels	UH is not providing software modules, so this does not apply.	N/A	N/A	N/A	Parameters and design guide lines	All CFs which address interaction.	X	X	X	X	X
	2	LAAS									X	X	X	X
RET	3	IPA	Under development	ToF depth sensor , color camera	Arm and gripper	Images from the sensors	Sensors, main control	Label, position and pose of a rigid object	LCT	Learning and operation	X	X	X	X
	1	EPFL									X	X	X	X
	3	UKA									X	X	X	X
MHP	2	LAAS									X	X	X	X
	2	LAAS	?	No direct sensor connection	Mobile platform Manipulator	Gestures Objects with position Sequence of human position Activities	GR OR PTA ACT	Motion commands	Platform Manipulator	?	X	X	X	X
	2	LAAS	?	No direct sensor connection	Mobile platform	Gestures Objects Sequence of human position Activities	GR OR PTA ACT	Motion commands	Platform	?		X	X	X
TBP	3	UKA	Fist implementations	3D – Sensors (stereo or Time of flight) optional + color cameras + laser range finder	Pan- tilt unit for 3D Sensors	List of 3d points some with assignments of body parts	Sensors	Human position Human Config- uration	GR, ACT, DLG or other HRI components		X	X	X	X
	3	UKA	Under construction			Sequence of human configurations plus object	TBP	Labeled activity and numerical parameters	DLG, execution components		X	X	X	X

CF/SF	KE	Partner(s)	CPU related				Real-time issues	Communication	OS	Language	Test Method
			CPU		RAM (HD)	Ports					
			Type	Usage							
DLG		UBI	P3, 500MHz	100%	256Mb	2 RS232	XCF (XML format)	Linux	C++	For some individual cogniron functions, test protocols may be suitable in the initial debugging phase.	
ROR		UBI	P3, 850MHz	100%	256Mb	2 RS232	XCF (XML format)	Linux	C++	For some individual cogniron functions, test protocols may be suitable in the initial debugging phase.	
PTA		UBI	PM, 1.7GHz	100%	512Mb	USB2.0	XCF (XML format)	Linux	C++	For some individual cogniron functions, test protocols may be suitable in the initial debugging phase.	
GR		UKA	P4, 2.6GHz	100%	1GB		CORBA	Linux	C++	As this function directly relies on CF-TBP and does not require much computational power for itself (compared to TBP), the requirements are more or less identical to CF-TBP.	
IA	1, 2, 3	UH	N/A	N/A	N/A	N/A	N/A	N/A	N/A	One test run with person A, another one with person B, personality profiles of both subjects are known. A difference in terms of the robot's behaviours towards the subjects could be demonstrated e.g. regarding social distances, how the robot attract's the subject's attention (which cues used) etc. Example 2: During the run the person grows uncomfortable with the robot, the robot realizes that and changes behaviour accordingly (we have developed a comfort measuring device but other partners by be able to contribute other recognition methods of discomfort – very hard though). This can be compared with a run where the robot is not able to detect subject's comfort level and thus does not change its behaviour	
LOC	1	KTH	2GHz	50%	100Mb	USB	Sockets	Linux	C++	The supervisor names locations in the learning mode. The robot can be tested by asking it for places.	
	1	UA	P4	100%	5GB	Ethernet	XCF	Linux	C++	We will make a number of image datasets with labelled ground-truth locations (room, kitchen,...). The localization-process can be split in two sub-tasks: the making of a higher order topological map of the environment and the localization of the robot in this map. Both these tasks can be evaluated separately by comparing the results with the ground-truth.	

		EPFL	P4, PowerPC	25% (real - time), 15% (non-real time)	1GB	USB FireWire PowerPC: RS422	10Hz obstacle avoidance, 1KHz motor control	Shared memory, Sockets, TCP/IP				
NAV	1	UA	P4	100%	5GB	Ethernet	Soft real-time 1Hz, 1s latency	XCF	Linux	C++		The robot should be able to go to a given location (for example "go to kitchen..."). The robustness to occlusions (for example when there are people around the robot) of the CF-NAV should be tested. The CF-NAV and CF-LOC should be integrated and tested in a number of test scenarios.
	1	EPFL										
SOC	1,2,3	UH	N/A	N/A	N/A	N/A	10Hz	N/A	N/A	N/A		User studies; see CF-1A
OR	1,3	IPA	P4, 2.0 GHz	100%	1GB	USB, FireWire	5Hz	None yet	Windows	C++		Annotated data bases can be used for testing. One such data base is the COIL data base, which can be found in the web. Additional databases need to be established for test of the pose estimation and which contain depth and color images. Such data bases are not available yet.
		LAAS										
		EPFL										
		UKA										
RET	2	LAAS										
MHP	2	LAAS										
NHP	2	LAAS										
TBP	3,2	UKA	P4, 2.6GHz	100%	1GB	FireWire Serial USB	10Hz	CORBA	Linux	C++		CPU usage depends also on the used tracking methods which serve as additional input to the 3d model based tracking. So CPU usage can be scaled with the required performance and accuracy. Anyway, 3d model fitting brings high computational costs.
ACT	3,2	UKA	P4, 2.6GHz	100%	1GB	Same as TBP	10Hz	CORBA	Linux	C++		As this function directly relies on CF-TBP and does not require much computational power for itself (compared to TBP), the requirements are more or less identical to CF-TBP.
LCT	3	UKA	P4, 2.6GHz	100%	1GB	Firewire 3xRS232	16Hz	CORBA	Linux	C++		Script 2 in KE3
RG	3	EPFL	P4	90%	1Gb	USB2.0 RS232	No real -time	Sockets between vision and control PC				See script 1 KE3
		UH	P4	-90%		USB2.0	Plans for	Sockets or	Linux			Demonstration in interaction with humans, e.g. Learning to set a

C.2 CF Implementation and Testing Status Overview

CF	Algorithm	KE	Setup	Contributor	State	Testing method(s)	Future (see the RA plans)
DLG	Many	1	Biron	UniBi	1st version Running	Script protocol	KE1, improvements from RA1
ROR	ROR	1	Biron	UniBi	1st version Running	Script protocol	KE1, improvements from RA1
PTA	PTA	1	Biron	UniBi	1st version Running	Script protocol	KE1 plus "active memory"
GR	Classification based on TBP	3	UniKarl, IPAbot	UniKarl	1st version Running at UniKarl and first port to IPA	Test data base published by UniKarl in the web	Full port to IPA, improvements, input to ACT, KE1 and KE2
LOC	Localization based on omni cam	1	Biron	UvA, EPFL	1st version Running at UvA and first port to UniBi	Test data base published by UvA in the web	Full port to UniBi, improvements resulting from RA5
NAV	Navigation	1	Biron	UvA, EPFL	1st version Running at UvA/EPFL and first port to UniBi	Test data base published by UvA in the web	Full port to IPA, improvements resulting from RA5
OR	Detector: SIFT/SVM	3	IPAbot	IPA	1st version Running at IPA and first port	COIL data base and derivations (contact IPA) and an internal data base	Improvements (RA5 plan), available to all partners
	Model maker: 3D Superquadrics	3	UniKarl	UniKarl	1st version Running at IPA and first port	Test data bases in preparation to compare the LAAS and UniKarl approach	Linkage to the feature points (IPA), see RA5 plan
	3D Mesh modelling	2	LAAS Jidowanki	LAAS	1st version Running at LAAS and first port	Test data bases in preparation to compare the LAAS and UniKarl approach	Hand-eye calibration
	Sensory-motor bindings	2	LAAS	LAAS	1st version Running at LAAS and first port	COIL data base and real-world examples	Estimate object rotation as "action"
MHP	Grasp planning in the human vicinity	2	LAAS Rackham	LAAS	1st version Running at LAAS and first port	Real-world tests for development. Data bases need to be defined	Improvements from RA6
NHP	Platform motion planning in the human vicinity	2	LAAS	LAAS	1st version Running at LAAS and first port	Real-world tests for development. Data bases need to be defined	Improvements from RA6
ACT	Activity detection based on TBP	3	UniKarl	UniKarl	First tests at UniKarl	Test data base (streams of range images) at UniKarl for some simple activities	Addition of activities
LCT	Task sequence learning	3	IPAbot	UniKarl	First implementation at UniKarl and activity converter at IPA	Test data bases with recordings of the human motion	See RA4 plan 2006
LJF	Task feature extraction	3	HOAP	EPFL	First version	Test methods from RA4 (metrics)	See RA4 plan 2006
RG	Imitation	3	HOAP	EPLL	First version	Test methods from RA4 (metrics)	See RA4 plan 2006

D Table of Software Modules

D.1 Glossary

Module: Name and quick description of the module

Services: List of the services included in the module

Tool/Library: Name and quick description of the tool or the library

Provider: Maintainer and affiliation. *Where does the software come from? Who can be contacted for more information?*

Environment: Required/supported OS, application, and/or framework. *Can this software be integrated into my existing system? For example, is it a module for OpenRobots? Is it a collection of Matlab scripts? Does it rely on Player/Stage²⁵?*

Bindings: Format of the API. *In what form (language) is the software implemented? In what form can it be used? For example, a library written C++ might have additional C, Python, and Java bindings. An all-binary application could use custom file formats for "bindings".*

Distribution: Downloadable packages, or links to external distribution pages. *Where do I get it? How can I install it?*

License: Conditions of use and distribution. *Can I legally use this software? Does it support my plans of commercializing or sharing the results?*

D.2 Software Available Through the Wiki

Tables 4 and 5 show the list of modules (groupings of services) and tools that were available on the Wiki in early 2006.

²⁵<http://playerstage.sourceforge.net/>

Table 4: List of shared software modules from the ModuleMarket page of the Wiki [14], as of early 2006. The meaning of the terms is explained in the glossary of this appendix.

Module	Services	Provider	Environment	Bindings
ROR : Resolving object references	GestureDetection, ObjectDetection, ObjectAttention	UniBi	XCF	C++, Java
HumanAct : Detecting human activities	RecoMotion, RecoAct	Steffen Knoop, UniKarl	Linux	C++ (some Matlab, C)
BodyModel : Model and track a human body	3DHumanModelFiller, PlausibilityChecker, Tracking	Jens Kubacki, IPA	todo	todo
ImitateWhatTo : Representation/encoding of human demonstrations	dimensionalityReduction, extractConstraints, generalizeSignals	Sylvain Calinon, EPFL	Linux/Windows	Matlab/Executable
ImitateMetric : Evaluation of a reproduced task	evaluateMetric	Sylvain Calinon, EPFL	Linux/Windows	Matlab/Executable
ImitateHowTo : Reproduction of a task in a different context	optimalMotionGeneration, forwardKinematics, inverseKinematics	Florent Guenter, EPFL	Linux/Windows	C/Executable
NHP : Navigation in Human Presence	NavPlanner, NavExec	E. Akin Sisbot, LAAS	OpenRobots	C, Tcl
MHP : Manipulation in Human Presence	ManipPlanner, ManipExec	E. Akin Sisbot, LAAS	OpenRobots	C, Tcl
twoDDetector : Human detection in images	FaceDetector, ColorBlobDetector, MotionDetector	M. Fontmarty, LAAS	OpenRobots	C, Tcl

Table 5: List of shared tools and libraries from the ModuleMarket page of the Wiki [14], as of early 2006. The meaning of the terms is explained in the glossary of this appendix.

Tool/Library	Provider	Environment	Bindings	Distribution	License
Sunflower: <i>Mobile robot abstraction, path planning, and obstacle avoidance</i>	Roland Philippsen, LAAS	Linux, OS X	C++, C	http://www.laas.fr/~rolo/sunflower.html	GPL, BSD
XCF: <i>Data- and event-driven system integration framework, utilizing XML technologies, middleware abstraction</i>	Sebastian Wrede, UniBi	Linux	C++, Java	http://xcf.sourceforge.net/	GPL
OpenRobots	Sara Fleury, LAAS	Linux	C, Tcl	http://softs.laas.fr/openrobots/ http://www.laas.fr/~rolo/openrobots-start.html	BSD
fakeloc: <i>Dummy implementation for CF-LOC</i>	Roland Philippsen, LAAS	Linux, OS X	C	http://www.laas.fr/~rolo/fakeloc.html	BSD
sflModule : <i>OpenRobots wrapper for Sunflower library</i>	Roland Philippsen, LAAS	OpenRobots (ie Linux)	C, Tcl	http://www.laas.fr/~rolo/sflModule.html	BSD
OmniCalib : <i>Calibration of omnidirectional cameras</i>	?DavideScaramuzza, EPFL	Matlab	Matlab	see OmniCalib page	GPL
HumanModelViewer	Stefan Vacek, UniKarl	todo	todo	http://www.iain.ira.uka.de/users/vacek/RA2/	todo

E Key Experiment Specification Details

This appendix contains the updates to the workplan of RA7 for phase 2 that resulted from the refinement process of the KE specifications.

E.1 KE1 Details

The following script was used for developing KE1 during phase 2. Table 6 lists what each step of the script implies in terms of robotic abilities, CFs, partner interactions, and environmental conditions. The latter are listed in table 7.

1. The robot (R) looks for human communication partner. It finds a person, directs its attention to it and, if necessary, tries to get the person's attention.
2. The person "registers" itself to R. R recognizes the person and subsequently keeps its focus on this partner. R uses person-dependent social spaces during the following interaction.
3. The person tells R the name of the current new location (kitchen).
4. The person asks R to follow him/her and R complies.
5. R follows the person until the person gives a new instruction.
6. The person points to various objects and tells R their names.
7. The person tells R the name of the current location (living room).
8. The person asks R to go back to the kitchen and R complies.
9. R goes to the kitchen.
10. As R arrives in the kitchen another partner can start a new interaction.

Table 6: Implications of the KE1 script

Step of Script	Abilities (robot)	Hardware (sensors)	Supporting Functions (SF)	Cogniron Functions (CF)	Environment Conditions (table 7)	Partner
1.	Human (body part) detection/tracking and attention control	laser sensor color cam stereo micros		PTA	1./2./5.	UniBi
	Getting attention of person			Parameters for PTA	Modalities for attracting attention derived from user studies	UH

continued on next page...

Table 6: ... continued Implications of the KE1 script

Step of Script	Abilities (robot)	Hardware (sensors)	Supporting Functions (SF)	Cogniron Functions (CF)	Environment Conditions (table 7)	Partner
2.	[See 1.] + Identify person and apply person-dependent social spaces	color cam stereo micros	Person Identification	Parameters for PTA	Personality types from user studies	UH
	Carry out dialog with user...	speech microphone	Speech recognition	DLG	2./6.	UniBi
	... using personalized dialogue strategies			Parameters for DLG	Styles of user interaction from user studies	KTH
3.	[See 2.] + Learn location	omni cam		LOC	3.(b)	UvA
4.	[See 2.]					
5.	[See 1.] + Follow person			PTA	5.(a)	UniBi
6.	[See 2.] + Learn object	stereo cam	Gesture and object recognition	ROR	4./5.(b)	UniBi
7.	[See 3.]					
8.	[See 2.] + Retrieve location		Scene model		6.	UniBi
9.	Navigation to location	odometry	Navigation	LOC	1./3.(a)	UvA
10.	[See 1.]					

Table 7: Environment specification for KE1

Environment Item	Feature	Conditions/Performance
PHYSICAL CONDITIONS	LIGHT (apply to all)	1. lighting of only one type (either daylight or artificial light)
	SOUND (apply all)	2. one person speaking, low background noise (office like, below 50 dB)

continued on next page...

Table 7: ...continued Environment specification for KE1

Environment Item	Feature	Conditions/Performance
ROOMS	<ul style="list-style-type: none"> ● hallway (lab-like environment with additional furniture/wallpaper/...) ● kitchen (simple structure, slightly more complex than lab environments) ● living room (more complex, more objects and different kind of furniture) 	<p>See 1./2. 3. (a) "easy" to navigate (sufficient distance between objects, all obstacles can be detected by robot's sensor, object - free space ratio), static or low velocity (0.5 m/s) dynamic objects (b) rooms are reliably (95 %) discriminable by robot's sensors</p>
LOCATIONS	<ul style="list-style-type: none"> ● hallway ● living room ● kitchenette ● dinette (dinner table area) ● working area (desk with laptop, shelf with books) ● lounge 	<p>See 3.</p>
FURNITURE	<ul style="list-style-type: none"> ● table ● chair / sofa ● sideboard, shelf ● micro-wave oven ● TV-set ● door ● window ● light switch (on wall) 	<p>See 3.</p>

continued on next page...

Table 7: ...continued Environment specification for KE1

Environment Item	Feature	Conditions/Performance
OBJECTS	<p>INSTANCES</p> <ul style="list-style-type: none"> ● book (standing in a shelf or laying on the table) ● plate ● coffee mug ● cellular phone ● laptop ● pen ● remote control ● coke can ● beer bottle ● banana ● orange ● plant (changes appearance over time!) 	4. (a) rigid objects, book closed, empty cup
	<p>PROPERTIES</p> <ul style="list-style-type: none"> ● color ● relations (here with examples, see relevant positions below): in (e.g., the shelf), on (e.g., the table), at (e.g., the door), in front of (e.g., the window), next to (the laptop) (left of, right of) 	4. (b) 10 colors.
	<p>POSITIONS</p> <ul style="list-style-type: none"> ● shelf (in/on/in front of) ● sideboard (in/on) ● dinner table (on,) ● office desk (on) ● floor (on) ● door (at/in front of/next to) ● window (at/in front of) ● worktop/kitchenette desk (on) 	4. (c) no occlusion, minimal distance 0.2 m

continued on next page...

Table 7: ...continued Environment specification for KE1

Environment Item	Feature	Conditions/Performance
HUMANS	ACTIVITIES <ul style="list-style-type: none"> • current looking direction (rough head orientation) • deictic gestures (fast movement, non-linear movement) • two people chatting 	5. (a) speed of movement <0.25m/s (b) distance of 0,60m-1,20m between robot and human (c) number of humans ≤ 4 (d) certain properties of the interaction partners are known (face, name, ...)
	DIALOGUE: <ul style="list-style-type: none"> • Handling simple instructions • Handling simple descriptions <ul style="list-style-type: none"> • object reference • object feature description • person reference • Handling queries for information retrieval (who, where, what) • Initiation of repairs • Handling socialization aspects (greeting, parting, apology, thanks) • Initiation of reports about robot's current state (especially in case of technical problems) • Self-explanation about name and capabilities 	6. language of restricted complexity, i.e., simple sentence structures with free word order (limited vocabulary ≈ 1000 words)

E.2 KE2 Details

The following script was used for developing KE2 during phase 2. Environmental conditions are listed in table 8.

1. The robot is observing an empty table top.
2. A human puts an unknown object on the table (box, cup).
3. The robot observes the object and moves its visual apparatus for that.
4. The robot computes grasping positions.
5. The robot holds the object for further exploration.
6. The robot prompts the human for naming the object.

7. The human gives a name and the robot repeats it.

Table 8: Environment specification for KE2

Environment Item	Feature	Conditions/Performance
PHYSICAL CONDITIONS	LIGHT (apply to all)	normal indoor lighting
	SOUND (apply all)	2. one person speaking, low background noise (office like)
ROOMS	Office room	
LOCATIONS	pre-specified working region: table-top.	
FURNITURE	table/desk	
OBJECTS	INSTANCES <ul style="list-style-type: none"> • box • soda can • coffee cup • free shapes 	rigid and manipulable objects, bottle closed, empty cup.
	PROPERTIES <ul style="list-style-type: none"> • Isolated objects 	
	POSITIONS <ul style="list-style-type: none"> • Object on table top 	
HUMANS	ACTIVITIES <ul style="list-style-type: none"> • Putting object on table • Manipulating object 	(a) speed of movement <0.25m/s (b) One human

E.3 KE3 Details

The following scripts (teaching and reproduction) were used for developing KE3 during phase 2. Table 9 lists what each step of the script implies in terms of robotic abilities, CFs, partner interactions, and environmental conditions. The latter are listed in table 10.

Teaching

1. The teacher attracts the attention of the robot by a specific speech command.
2. The robot moves to the direction of the teacher to enable itself for receiving commands.
3. The teacher gives the robot a command by a selected gesture or by speech that he/she wants to teach the robot a new pick&place task.
4. The teacher takes an object from the shelf and shows it to the robot.

5. The robot detects the object (for classification) and the hand (for tracking).
6. The teacher moves the object onto the desired location on the table.
7. The robot tracks the human's arm until it is disconnected to the object and stores its position.
8. The teacher takes the next object and steps 4) to 7) are repeated until a small set-up is finished.
9. The teacher expresses the end of the task through a selected gesture or speech command.
10. The robot stores the set-up information.

Reproduction

11. The teacher attracts the attention of the robot by a specific speech command.
12. The robot moves to the direction of the teacher to enable itself for receiving commands.
13. The teacher gives the robot a command by a selected gesture or by speech that the robot has to set-up the configuration learned in steps 1) to 10).
14. The robot reproduces the setting by chaining (a) pick and (b) place operations.

Table 9: Implications of the KE3 script

Step of Script	Abilities (robot)	Hardware (sensors)	Supporting Functions (SF)	Cogniron Functions (CF)	Environment Conditions (table 10)	Partner
1.	Speech recognition of selected commands	Microphones	Speech recognition (external)		2./7.	UniBi
	Gesture recognition of selected commands	TOF sensor color cam		GR	1./7.	IPA
	Human (body part) detection and tracking	TOF sensor color cam		TBP	1./6.	UniBi
2.	Platform movement	Laser scanner mobile platform	Indoor navigation		6. (b)	IPA
4.	Hand detection	TOF sensor color cam		TBP	6. (a)	UniKarl
	Object detection/classification	TOF sensor color cam		OR	1./5.	IPA

continued on next page...

Table 9: ... continued Implications of the KE3 script

Step of Script	Abilities (robot)	Hardware (sensors)	Supporting Functions (SF)	Cogniron Functions (CF)	Environment Conditions (table 10)	Partner
7.	Arm/hand tracking	TOF sensor color cam		TBP	1./6 (a)	UniKarl
9. 11. 13.	See 1.					
12.	See 2.					
14. (a)	Object pose estimation	TOF sensor color cam		OR	5.	IPA
	Collision free object grasping	TOF sensor color cam arm	Grasp planning & execution		1./5.	IPA
14. (b)	Collision free object placement	TOF sensor color cam arm ft sensor	Place planning & execution		1. + free target area available	IPA

Table 10: Environment specification for KE3

Environment Item	Feature	Conditions/Performance
PHYSICAL CONDITIONS	LIGHT (apply to all)	1. normal indoor lightening, static
	SOUND (apply all)	2. one person speaking, low background noise (office like, <50dB)
ROOMS	living room like lab setting	see 1./2.
LOCATIONS	<ul style="list-style-type: none"> • dinette (dinner table area) • pre-specified working regions: sideboard, shelf. 	3. Normal heights (<1.5 m)
FURNITURE	<ul style="list-style-type: none"> • table • chair / sofa • sideboard, shelf 	See 3.

continued on next page...

Table 10: ... *continued* Environment specification for KE3

Environment Item	Feature	Conditions/Performance
OBJECTS	INSTANCES <ul style="list-style-type: none"> ● book (standing in a shelf or laying on the table) ● plate ● box of biscuits ● bottle of water ● glass ● coffee mug 	4. rigid objects, book closed, bottle closed, colored glass, empty cup.
	PROPERTIES <ul style="list-style-type: none"> ● color ● weight ● grasping points ● functional roles 	5. all colors, partial occlusions <15%
	POSITIONS <ul style="list-style-type: none"> ● shelf (in/on/in front of) ● sideboard (in/on) ● dinner table (on, relative to) 	5. pre-specified with variances <0.3m
HUMANS	ACTIVITIES <ul style="list-style-type: none"> ● deictic gestures (fast movement, non-linear movement) ● object manipulation (carrying something, answering phone, ...) 	6. (a) speed of movement <0.25m/s (b) number of humans ≤2
	DIALOGUE <ul style="list-style-type: none"> ● selected commands to activate the implemented functionalities by speech recognition ● selected commands by gesture recognition 	7. selected set of keywords and gestures (<15)