ICT Call 7
ROBOHOW.COG
FP7-ICT-288533

Deliverable D7.4:

Architecture design with basic components integration

January 31st, 2013

Project acronym:	ROBOHOW.COG
Project full title:	Web-enabled and Experience-based Cognitive Robots that Learn Complex Everyday Manipulation Tasks
Work Package:	WP 7
Document number:	D7.4
Document title:	Architecture design with basic components integration
Version:	1.0
Delivery date:	January 31st, 2013
Nature:	Report
Dissemination level:	Public
Authors:	Moritz Tenorth (UNIHB) François Keith (CNRS)

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement n^{o}288533 ROBOHOW.COG.

Contents

Chapter 1
Introduction

In the following chapters, we will recapitulate the work packages in RoboHow and then explain the planned RoboHow architecture from different points of view. First, we will outline the main functional components of the system, their connections and relations to the work packages. Then, we will describe the different representations in RoboHow and the flow of information between them. Finally, we will describe three use cases in which sub-groups of the components work together: the analysis and formal representation of observations of human actions, imitation learning and execution on the robot, and the import of natural-language instructions into the knowledge base.

Chapter 2
Summary of the work-packages

WP 1 "Representation" designs, develops, implements, and uses multiple representations of everyday manipulation tasks, investigates how these representations can be interlinked and even combined into a hybrid representation, and explores how robots can reason with these representations.

WP 2 "Observation of Human Demonstrations" provides the symbolic representation of demonstration videos that are required to learn models of manipulation activities and translate them into the robotic platforms manipulation strategies.

WP 3 "Constraint- and Optimization-based Control" develops an action and movement interpretation system that generates fast, smooth, and dynamically adequate movements for the constraint- and optimization-based action specifications.

WP 4 "Perception for Robot Action and Manipulation" enables object detection and manipulation in natural settings based on single and multi-sensory feedback in an inside-out manner, retrieving knowledge about the environment and the functionality of objects and their attributes using the robot's sensory capabilities.

WP 5 "Learning from Interaction with a Human" aims at providing the robot with the ability to learn how to perform manipulatory tasks from and when interacting with a human. Learning will proceed in an incremental manner by bootstrapping the system with observations of large sets of manipulatory activities, that will later guide and speed up the learning of each manipulatory task.

WP 6 "Plan-based Control" designs, implements, and analyzes a new generation of plan languages and computational models for plan-based robot control including issues like plan representation, prediction mechanisms, transformational planning and learning, cognitive models of everyday manipulation, and flexible execution monitoring.

Chapter 3

RoboHow architecture and functional components

Figure 3.1 shows the main functional components of the RoboHow system. The CRAM system (WP6) thereby serves as an "interface layer" that supervises and orchestrates the lower-level control components, triggers perception routines, interprets their results, and altogether provides the basis for executing the abstractly specified plans generated from Web instructions. These plans make use of different kinds of information stored in the robot's knowledge base that is complemented with information from sources on the Internet (WP1) and from observations of humans (WP2).

During task execution, the plan-based controller acquires object information from the perception (WP4) and executes the plans by sending constraint-based motion specifications to the lower-level controllers (WP3). In order to obtain smooth motion, single motion segments can be optimized and blended. The imitation learning module (WP5) learns motions from observations of human actions and thereby provides information to the controller on how the motions are to be performed. All components operate on top of existing low-level control routines and perception methods. Those components are required for successfully executing the tasks on the robot, but are beyond the scope of the RoboHow project. They are partly already available from prior work and are partly being developed in parallel satellite projects (e.g. the EU FP7 project SAPHARI for human-safe motion control (UNIHB) or the EU FP7 project ROSETTA (KUL)).

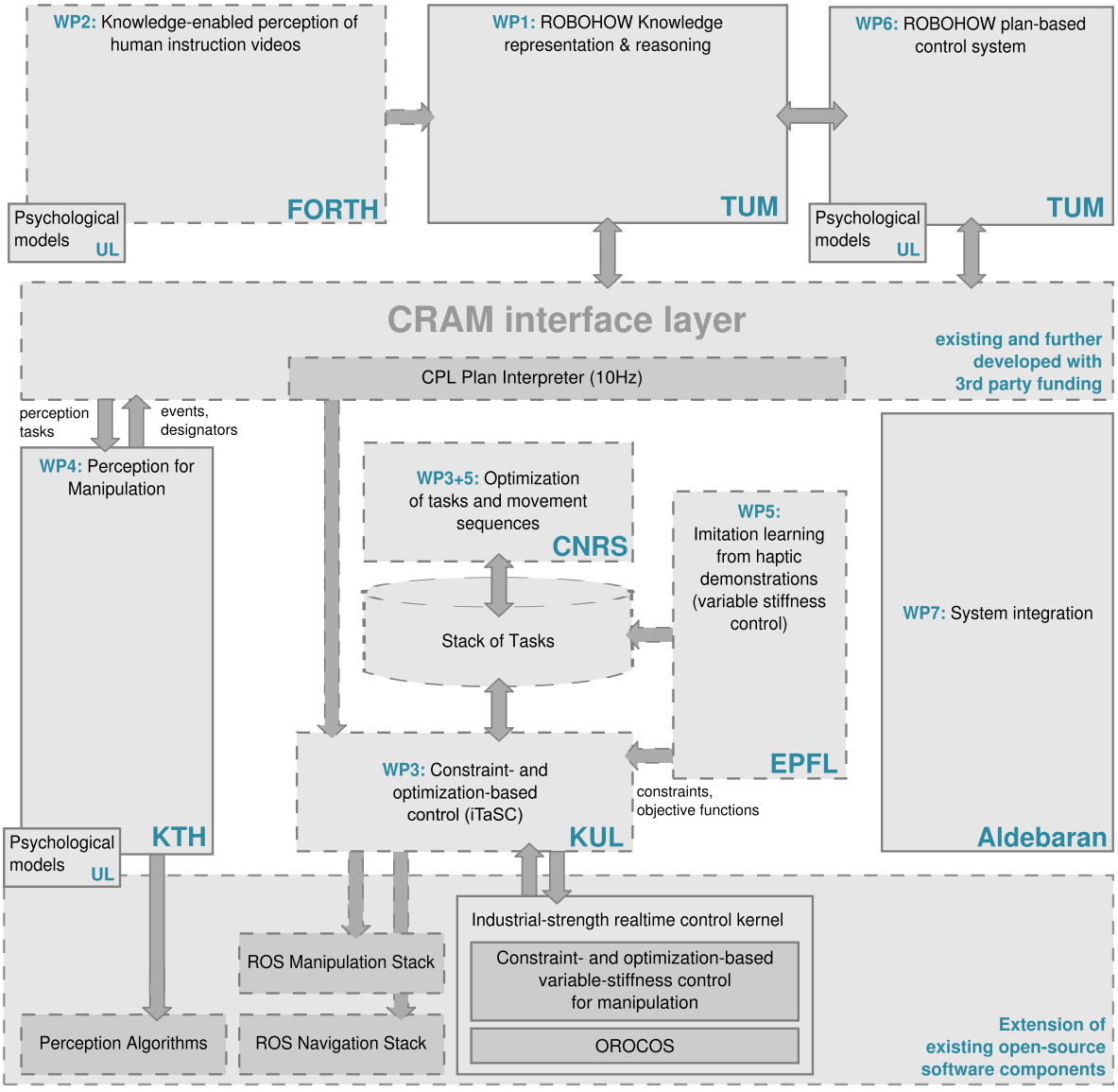

Figure 3.1: Functional overview of the RoboHow system including references to the work packages and partner institutions providing this functionality.

Chapter 4

Flow of knowledge between the RoboHow components

A central aspect of the RoboHow project is the representation, processing, and integration of various sources of information in different modalities. In this chapter, we will take a closer look at what kinds of information exist, how they are represented, where they are acquired from, and how they are used in the context of executing everyday manipulation tasks.

Figure 4.1 outlines the relation between on the one hand the three main representations developed as part of WP1, the AbstractEvaRep, the HumanEvaRep, and the RobotEvaRep, and on the other hand the execution components. The reason that these different representations exist is that there are different ways how an action can be described:

- The AbstractEvaRep contains abstract information about the structure of the action itself, independent of how or by whom it is executed. This information can be partially obtained from Web instructions that specify the main steps to be taken to execute a task, but do not contain any information about grasps, forces, trajectories, motions, object coordinates or similar. In the context of RoboHow, the AbstractEvaRep provides the general vocabulary for describing actions and forms the link between the other representations. The AbstractEvaRep describes classes of actions and their properties.

- The HumanEvaRep is a formal representation of observations of human activities. Its elements are instances of the abstract action classes defined in the AbstractEvaRep, i.e. actions that have been carried out at some point in time. Therefore, this representation can provide all kinds of information that have been observed, including timing, coordinates, trajectories, etc. in addition to knowledge that is inherited from action classes in the AbstractEvaRep via the class–instance relation. Compared to the AbstractEvaRep, the HumanEvaRep is therefore more embodied, grounded and situated. It does, however, only describe sets of specific task executions, so applying the contained information to new situations requires the system to abstract it and to draw the connection between the situation at hand and similar situations in the HumanEvaRep.

- Finally, the RobotEvaRep contains action-related information that is needed for executing the tasks on a robot. This information is also described on a class level, but in addition to the AbstractEvaRep, it contains e.g. links to object models that can be used for recognition,

motion specifications that describe how sub-actions can be performed etc. In many cases, the RobotEvaRep will not directly describe complete tasks, but rather the "building blocks" from which complex tasks can be composed (that are described in the AbstractEvaRep). This modular description of knowledge is expected to improve generalization across tasks.

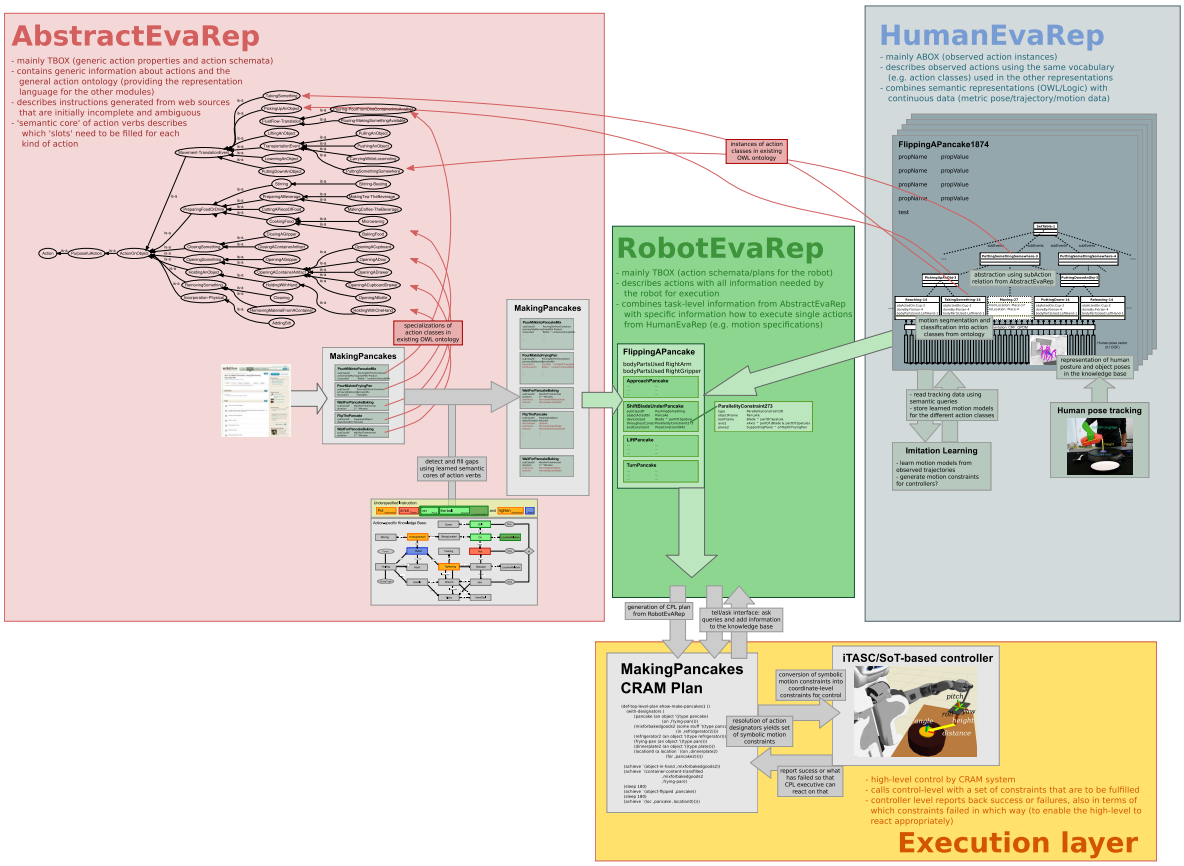

Figure 4.1: Overview of the knowledge flow in RoboHow. The picture contains information at several scales and is therefore best viewed on a screen so that one can zoom into the different parts. Each block is also presented in a larger scale on the following pages.

4.1 AbstractEvaRep

This section gives an overview of the main components of the abstract action representation AbstractEvaRep (Figure 4.2). An important part of this kind of knowledge is the ontology of action classes (depicted in the upper left of Figure 4.2)) that describes a hierarchy of actions and their respective properties. Properties of more generic classes are inherited by specialized classes derived from them. For example, one could specify in general that all object manipulation actions (*ActionOnObject*) require a model that can be used to recognize the manipulated object, and this rules automatically applies to all derived action classes.

The ontology provides the vocabulary for describing actions in the system. A concrete task, for example the task for making pancakes, can be described by deriving specialized action classes from the general classes in the ontology (e.g. pouring pancake mix into a frying pan as a special

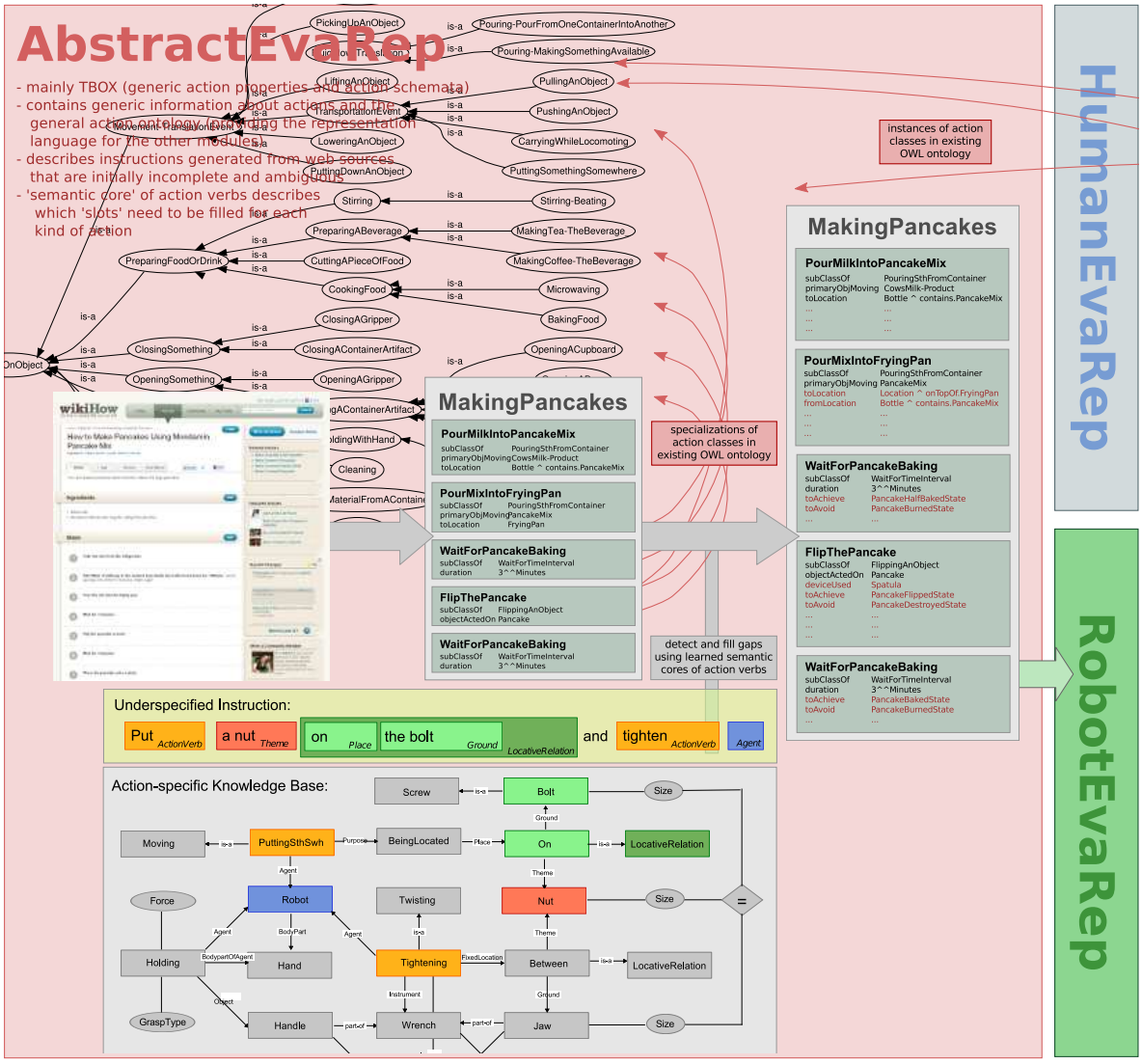

Figure 4.2: Abstract representation of actions in the ontology and in terms of class-level task descriptions generated from natural-language instructions. The center shows the path from the web instructions to a first formally-represented specification that describes the action in terms of the classes in the ontology (upper left). This initial representation is complemented using learned action-verb specific knowledge bases (bottom) with information that is not mentioned in the instructions themselves.

case of pouring liquid into a container), and these specialized action classes can be composed to a sequence or a more complex task structure. The blocks in the center describe how such a sequence of action classes can automatically be generated from natural-language instructions on the Web [Tenorth et al., 2010].
Web instructions are normally incomplete and lack information that their authors omitted because they considered it to be obvious. These knowledge gaps need to be detected and should be filled as good as possible using the robot's knowledge about action verbs and their arguments [Nyga and Beetz, 2012]. This is done using the action-verb specific knowledge bases developed in WP1 that apply generic "templates" of the concepts related to an action verb that have been learned from data to the instructions at hand to infer missing information. The result is an abstract task specification that is as complete as possible given the robot's knowledge base.

4.2 HumanEvaRep

The HumanEvaRep (Figure 4.3) is a formal representation of observed human activities in terms of the action classes defined in the AbstractEvaRep. Since the actions are described as instances of the abstract classes, it is possible to apply knowledge from the AbstractEvaRep to them. The HumanEvaRep is populated with observations from the human pose tracking system developed in WP2.
While parts of the HumanEvaRep will be on the symbolic level, these representations are grounded in the continuously-valued action observations generated by the tracking system. The translation between these is done by motion segmentation and classification methods. Combined, the two levels provide a comprehensive knowledge source about human actions that allows symbolic queries that return continuously-valued results. This part of the system is similar to the AM-EvA models described in [Beetz et al., 2010].
The HumanEvaRep will serve as information source for the imitation learning (WP5) modules by providing structured access to the human tracking data (e.g. for selecting parts of the motions based on semantic properties), but will also store the motion models learned as result of the imitation learning.

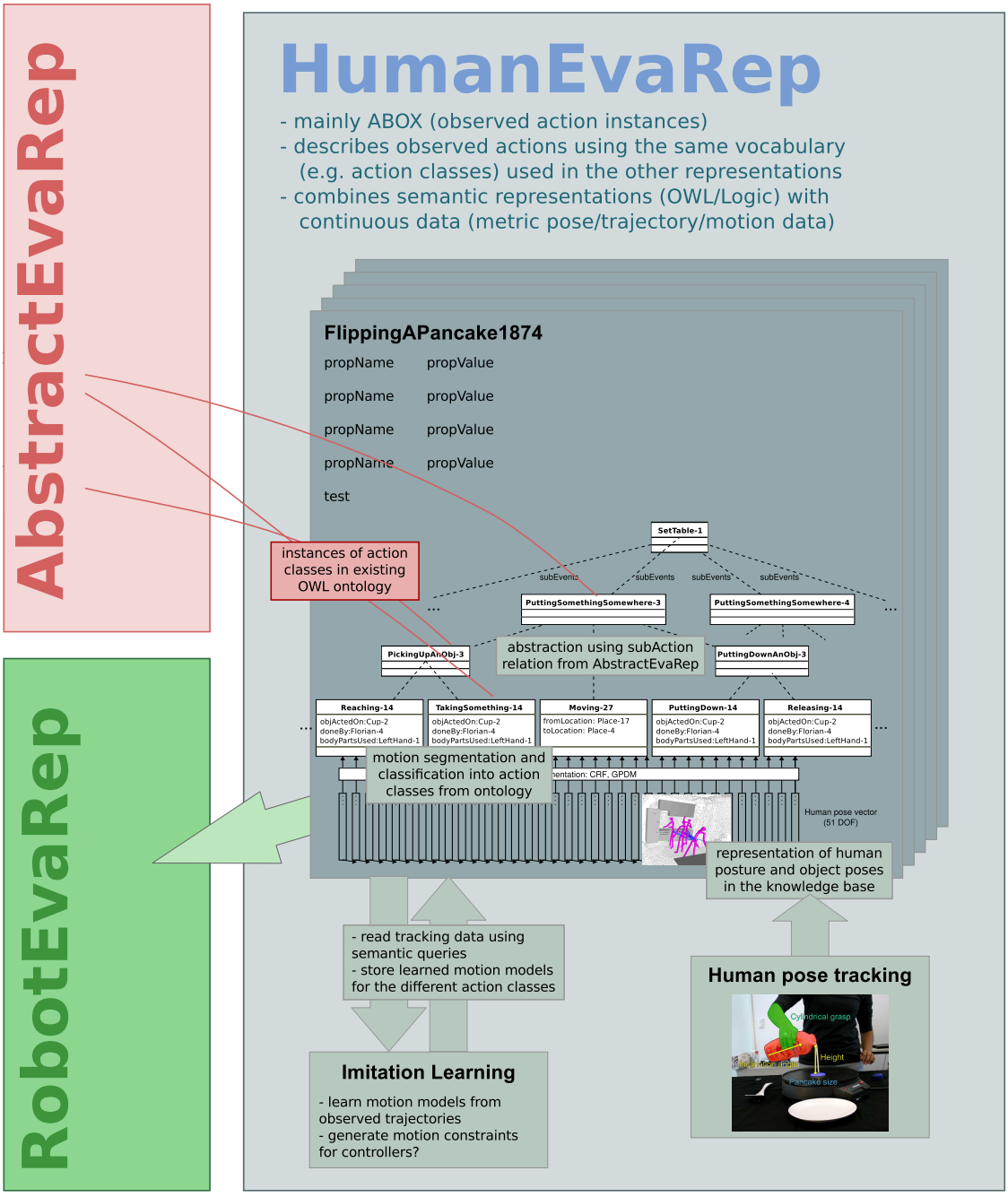

Figure 4.3: Combined symbolic and continuously-valued representation of observed human actions in the HumanEvaRep.

4.3 RobotEvaRep

The RobotEvaRep (Figure 4.4) combines information about abstract action properties and class-level descriptions of the structure and composition of tasks (from the AbstractEvaRep) with knowledge that has been extracted from human observations (in the HumanEvaRep) and creates robot-specific descriptions of the actions.
The overall task structure is obtained from the Web instructions that have been augmented with action-verb specific knowledge. What the Web instructions do not contain is information about the motions that need to be performed to execute each of these action blocks. This is the place where models learned from humans can provide important and complementary information. The imitation learning procedure interprets the observations of human actions and translates them into a format the robot can use – in RoboHow, these are constraint-based motion specifications that can be interpreted by the controllers described in the following section.

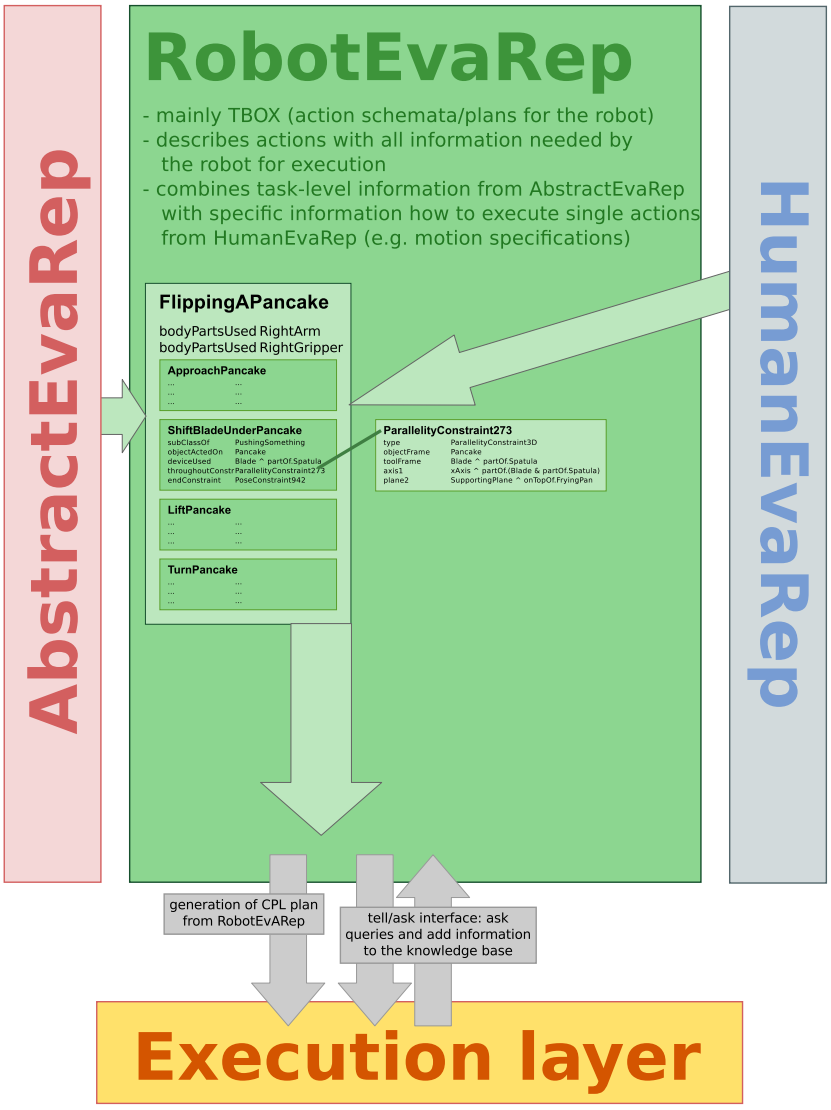

Figure 4.4: Robot-specific representation of task-related information in the RobotEvaRep.

4.4 Execution component

Before execution, the RobotEvaRep is translated into a robot plan in the CRAM Plan Language (CPL) that, in addition, provides control structures for task monitoring and supervision, integration with perception modules, concurrent execution etc. In CPL, motions are described in terms of action designators, which are partial symbolic motion descriptions that list a set of constraints that need to be obeyed during execution. The symbolic constraint specifications are translated into constraints on a geometric level and are passed to the constraint- and optimization-based controllers (WP3).

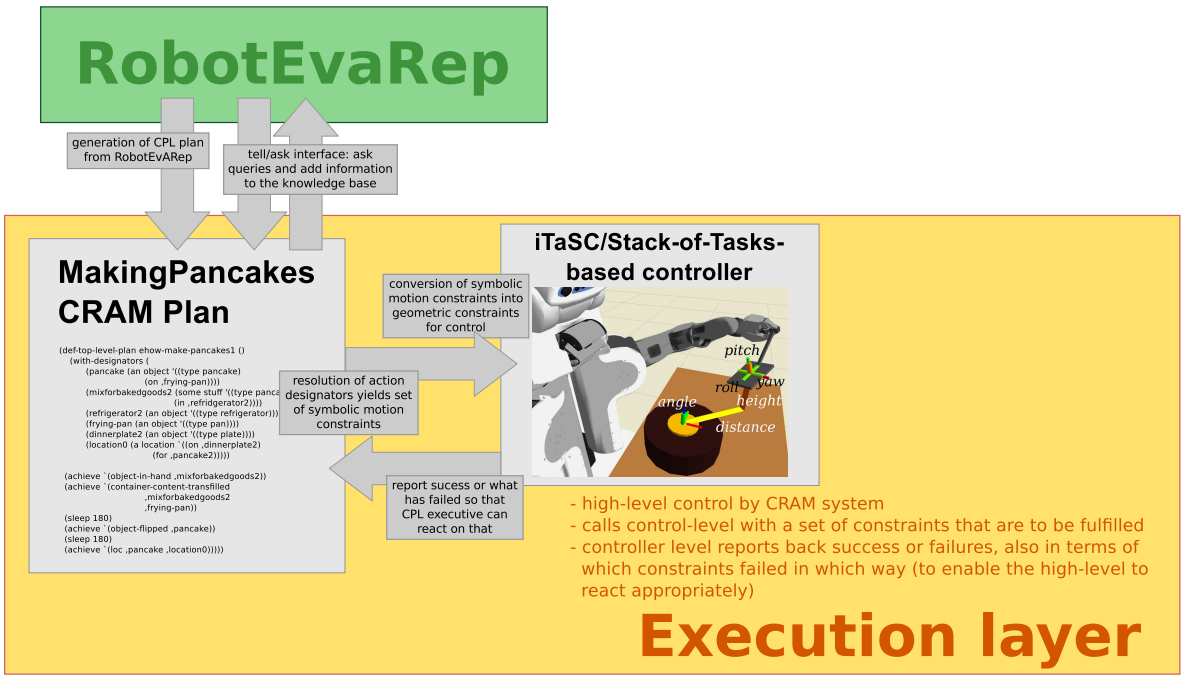

Figure 4.5: The execution on the robot includes task-level coordination as well as motion-level control that both use constraints as common interchange format.

Chapter 5
RoboHow use cases

In this chapter, we will give an overview of different use cases of the RoboHow system for information acquisition and task execution. The first two use cases describe the flow of information from knowledge sources into the internal representation, namely for observations of human actions and task instructions from the Web. The other two use cases describe the execution of action plans on the robot using the constraint- and optimization-based control framework and the learning of motion models by imitation.

5.1 Acquisition of task instructions from the Web

Figure 5.1 sketches the methods for the acquisition of knowledge from selected web sites such as *wikihow.com* and *ehow.com*. The natural-language instructions are first transformed into a logical representation using the Web importer presented in [Tenorth et al., 2010]. This abstract and incomplete representation is then complemented with additional knowledge and the learned probabilistic action cores developed in WP1 [Nyga and Beetz, 2012] and stored in the knowledge base.
The resulting specifications combine single actions into complete tasks described at an abstract level. They form the skeleton for the robot plans that are finally executed on the robot.

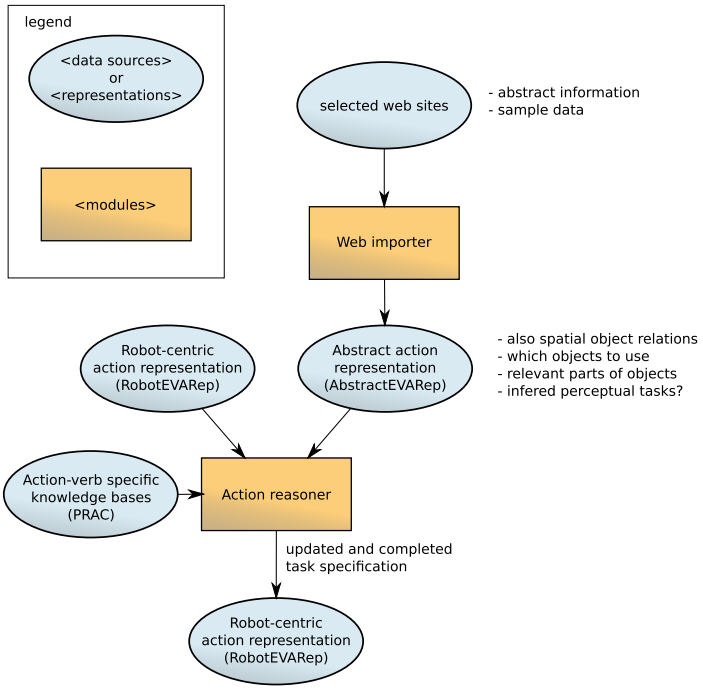

Figure 5.1: Knowledge acquisition by translating Web instructions into formal task specifications and filling information gaps using action-verb specific knowledge bases.

5.2 Observations of human actions as knowledge resource

While the web instructions provide a semantically rich and abstract task-level description, they lack information on the motion level. In order to execute them on the robot, these information gaps need to be filled and the descriptions need to be combined with detailed descriptions of how to perform the different motions. This kind of information can be obtained from observations of human actions that provide in a way the opposite of the information contained in Web instructions: the observations are detailed, embodied, situated in a specific environment, continuously-valued, but a priori unstructured and not semantically annotated.

To extract knowledge from observations of human actions (Figure 5.2), the system computes hand postures and object poses from the observed videos and tracks the motions over time. Both streams of information (objects and human motions) are aligned and can be segmented e.g. using information about contacts between the hands and objects or among objects. At different stages during the processing, the system can make use of higher-level knowledge about the actions to guide the tracking and interpretation, for example to select objects of interest or expected object interactions.

At least the initial version of the RoboHow system will concentrate on intentional task demonstrations in which humans perform all actions in a way that they can be easily observed and transferred to the robot. While other body parts can also perform important motions, like closing a drawer with the elbow, and while such actions could be described in the internal representations, the observation and execution components will first deal with manipulation tasks using the hands or grippers.

The result of the tracking and interpretation procedure is a combined continuous/discrete representation of the observed activities that contains on the one hand the detailed motions, postures, timing information etc, but that is also semantically annotated, linked to the action classes in the robot's knowledge base, and can be used for logical inference.

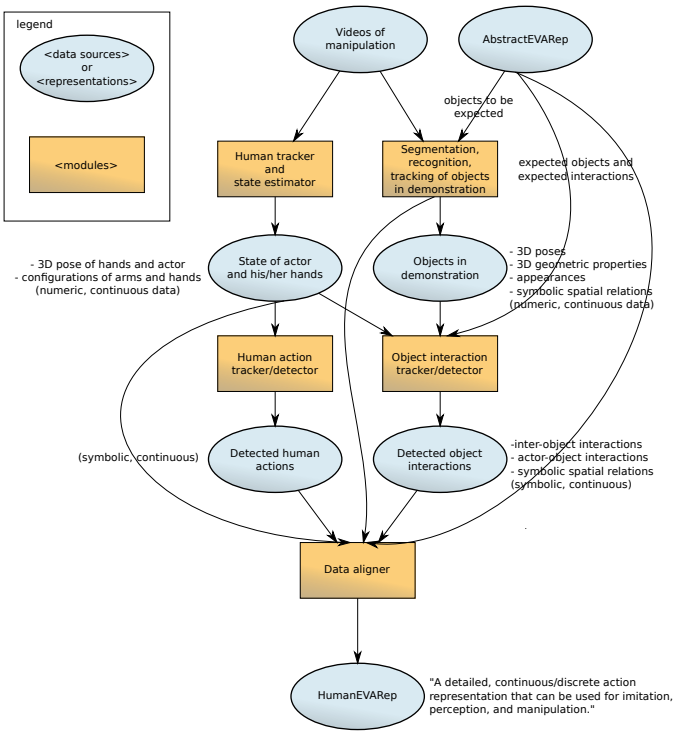

Figure 5.2: Knowledge acquisition by observing and interpreting human manipulation actions.

5.3 Imitation learning from observed manipulation actions

The third use case deals with the problem of how to combine the abstract action knowledge from the Web with the observed human demonstrations to learn action models that a robot can use. We expect that this procedure can be performed in three main phases (Figure 5.3): In a first step, the system applies imitation learning techniques to the observations obtained from the HumanEvaRep to learn an initial set of motion constraints that are stored in the RobotEvaRep (Task 1). Once these constraints have been created, the robot can start executing the motions itself, refine them, and adapt them to its own embodiment (Task 2). The updated specifications are again stored in the RobotEvaRep that, over time, contains a more and more comprehensive knowledge base about actions that is tailored to the respective robot. The learned constraints will be stored in a format that can be interpreted by the constraint- and optimization-based control framework. By aligning the learned motion models with the abstract task specifications (Task 3), the robot can associate sets of motion constraints with symbolically-described action classes. This will allow the plan-based controller to execute novel combinations of actions (i.e. novel tasks that are described in the AbstractEvaRep) using the learned models on the constraint- and optimization-based controllers.

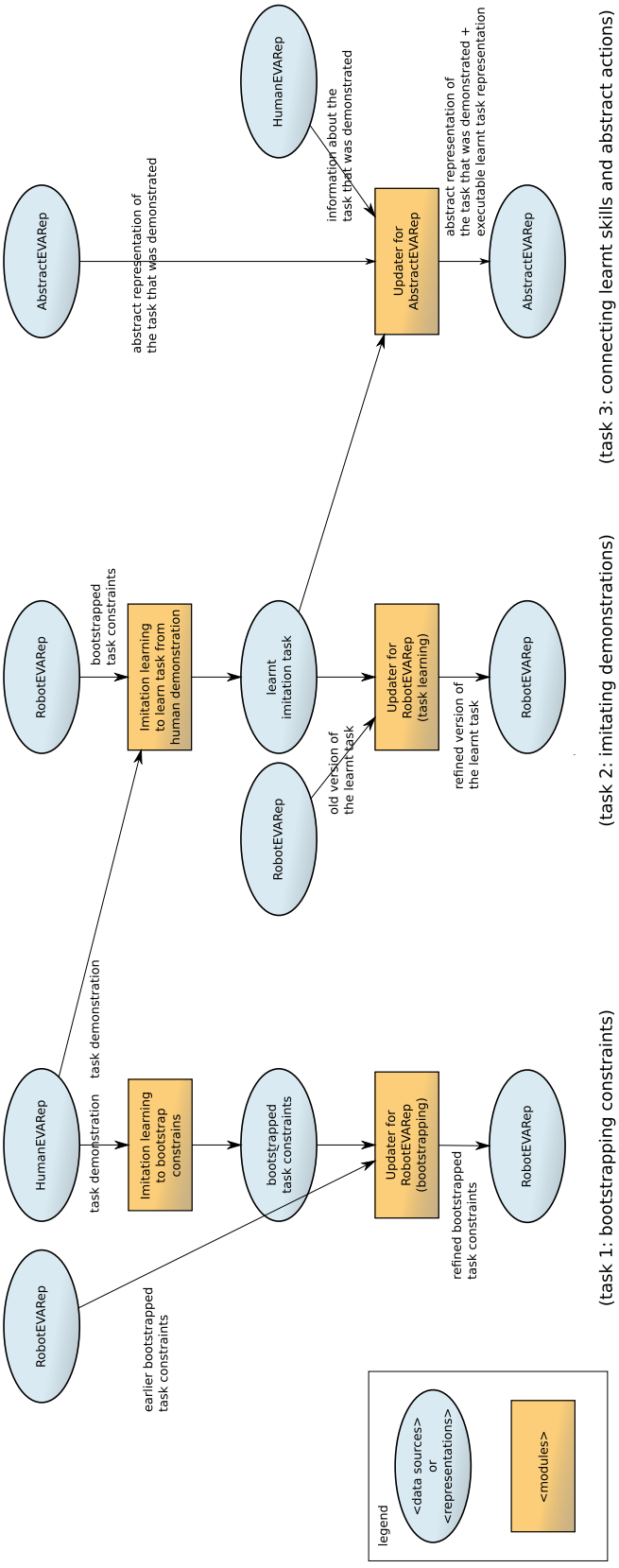

Figure 5.3: Imitation learning for translating from observed human actions to set of motion constraints that can be used in the robot's constraint- and optimization-based control framework.

5.4 Task execution in the constraint- and optimization-based framework

As a result of the imitation-learning procedure described in the previous section, the robot has learned sets of motion constraints describing how single actions have to be executed. These motion constraints are linked to the abstract representations of these actions in the knowledge base and form the "building blocks" for robot tasks.
The instructions from the Web describe which actions need to be composed in which way to perform a composite task such as making pancakes. The learned action models are composed to robot plans according to the structure extracted from these instructions that are described in the CRAM Plan Language (CPL). The plan-based executive reads these plans, sends the constraints to the motion controllers, triggers perception for monitoring and closed-loop manipulation, and handles observed execution failures.

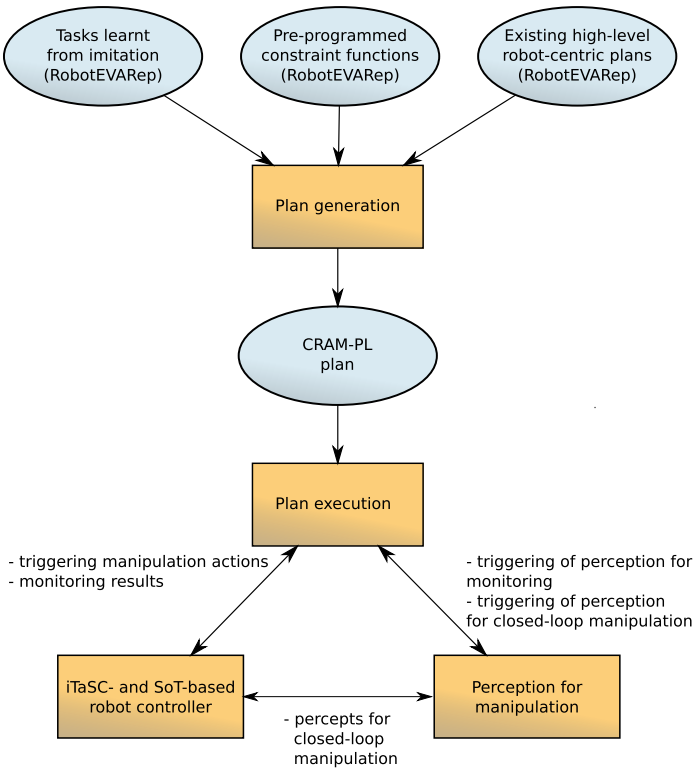

Figure 5.4: Execution of tasks by combining information about the structure of tasks obtained from the Web with learned motion constraints.

Chapter 6
Conclusions

In this document, we presented the RoboHow system architecture from different points of view. In Chapter 3, we introduced the main components of the system and their respective functions and tasks in the context of the overall system. Chapter 4 described the different knowledge representations that are used in the system and that need to be combined to achieve the overall goal of the RoboHow project of robots that are able to learn novel tasks by combining web instructions with experience knowledge. In Chapter 5, we sketched different use cases of the system for knowledge acquisition, imitation learning and task execution.
We expect this architecture design to be a living document that will be updated and adapted to novel developments over the course of the project. While the overall system architecture is to remain stable, the implementation of the functionality and the interfaces between the components will be defined in more detail while the components are developed, integrated into complete robotic systems, and evaluated on different tasks.

Bibliography

[Beetz et al., 2010] Beetz, M., Tenorth, M., Jain, D., and Bandouch, J. (2010). Towards Automated Models of Activities of Daily Life. *Technology and Disability*, 22(1-2):27–40.

[Nyga and Beetz, 2012] Nyga, D. and Beetz, M. (2012). Everything robots always wanted to know about housework (but were afraid to ask). In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal.

[Tenorth et al., 2010] Tenorth, M., Nyga, D., and Beetz, M. (2010). Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1486–1491, Anchorage, AK, USA.