Agile Acceptance of Domain Expert's Feedback with Developing Robot Tea House

Takahira Yamaguchi **Takeshi Morita**

Faculty of Science and Technology
Keio University

Abstract

We describe PRINTEPS with two facilities: AI integration and agile process. The former means combining four types of AI modules: knowledge-based reasoning, speech dialog, image sensing, manipulation & motion planning. The latter means incremental AI apps development process to smartly accept domain expert's feedback. We explain how PRINTEPS is going with developing robot tea house and evaluate how much well go AI integration and how much agile go the feedback from domain experts (no IT/AI experts).

1 Introduction

While AI applications are very popular in many domains, they should be harmonious with domain exerts and end users. Furthermore, in order to develop such harmonious AI applications, we need agile AI platforms for not only developers but domain experts. Such AI platforms should have AI integration and agile process facilities. The former means combining four types of AI modules such as knowledge-based reasoning, speech dialog, image sensing, manipulation & motion planning. The latter means incremental AI apps development process to smartly accept domain expert's feedback.

From the above background, we develop PRINTEPS (PRactical INTEligent aPplicationS), which is a user-centric platform to develop integrated intelligent applications only by combining four types of modules: knowledge-based reasoning, speech dialog, image sensing and motion management. PRINTEPS supports end users to care about AI application design (user participation design) and to develop and revise it easily. This paper describes PRINTEPS architecture and evaluate how much well go AI integration and how much agile go the feedback from domain experts (no IT/AI experts), developing robot teahouse applications with PRINTEPS, where multiple people and robots cooperate.

2 The Architecture of PRINTEPS

2.1 Overview

Figure1 shows the architecture of PRINTEPS. PRINTEPS is based on Robot Operating System (ROS), which is a robot framework and offers communication libraries and various tools. In ROS, a user can employ services, which have a synchronous communication mechanism, and topics, which offers a mechanism for asynchronous communications. Because ROS is implemented as a distributed system, its use thereof facilitates the handling of multiple robots or multiple sensors, and the use of topics enables various other processing operations, including the real time acquisition of sensor values. The input-output data type of services and topics, known as messages, can be defined as a composite data type such as a class or structure. Regarding services, the input-output data specifications can be defined, without relying on a programming language.

PRINTEPS mainly consists of 5 types of intelligent sub systems (knowledge-based reasoning, spoken dialogue

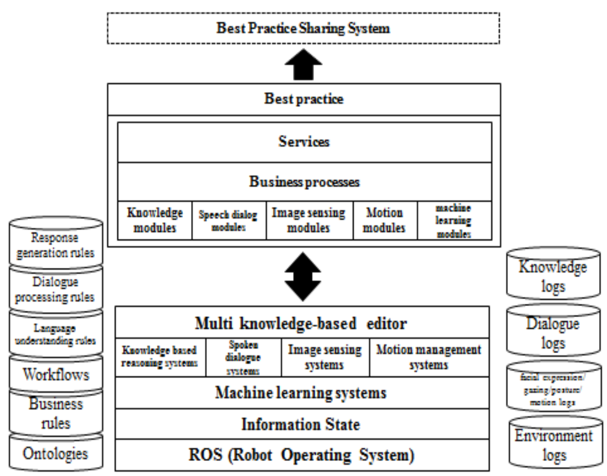

Figure 1 PRINTESP Architecture

understanding, image sensing, motion management and machine learning). Knowledge-based reasoning systems include workflows, business rules, and ontologies. Spoken dialogue understanding systems include response generation rules, dialogue processing rules, and language understanding rules.

Information State (IS) in PRINTEPS is a database for sharing data among intelligent sub systems. We use MongoDB_store , which is a MongoDB-based storage and analysis for data from a ROS system, as IS. In PRINTEPS, each sub system stores data to IS and refers data from IS. The data format in PRINTEPS is the format of messages in ROS.

In addition, knowledge logs, dialogue logs, facial expression/gazing/posture/motion logs, and environment logs obtained by humans and machines' multimodal interaction become input for batch and online machine learning.

PRINTEPS also provides a multi knowledge-based editor for end users to develop integrated intelligent applications easily by combining software modules from PRINTEPS. The details of the editor are described in the next section.

2.2 Multi Knowledge-based Editor

The multi knowledge-based editor consists of a workflow editor and a Business Rule Management System (BRMS).

Workflow Editor

Users can develop workflows using components based on Service Oriented Architecture (SOA) by the workflow editor. Main components in the workflow editor are modules, processes, and services. The modules are primitive functions and they are fully compliant with ROS services, topics, and messages. The processes play a function-like role in which they deal with multiple modules. In the process operation, processes and modules can be defined in a mixed manner. The services are functions with the coarsest grain size and are composed only of processes defined as business processes. The first hierarchy (route service) of the workflow editor is composed of a route start, service list, and route end. The service list, when completed, will be shared among users as a case library (best practice). It will allow users to locate easily those services, processes, or modules used as best practice.

The workflow editor automatically generates source codes in Python that can be executed in ROS.

The workflow editor also has two views. One is a workflow view which represents a sequence of the components. Figure2 shows a screenshot of workflow view in the workflow editor. The other is a scenario view which also represents a sequence of the components with actors.

Business Rule Management System

BRMS includes a Business Rules Engine (BRE), a web authoring and rule management application. We use drools as the BRMS. Since BRMS has a function to define rules with domain specific language, users can easily describe business rules with natural-language like form.

In PRINTEPS, “fire business rules” module in the workflow editor can execute business rules in the BRMS and the BRMS can execute modules in the “then-part” of

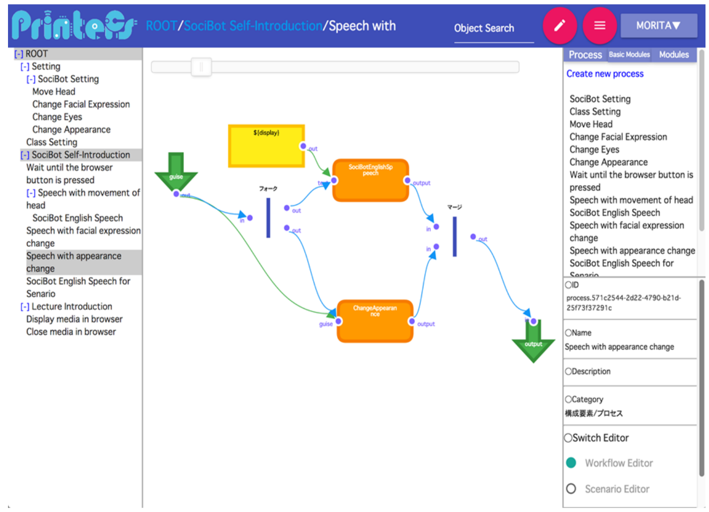

Figure2 A Screenshot of Workflow Editor

business rules. The big difference between a traditional knowledge-based system and an integrated intelligent application is that sensing results such as person attributes from an image can be described in the “when-part” of business rules and robots action such as speaking and moving to a certain place can be described in the “then-part” of the business rules.

We also use Semantic Web Rule Language (SWRL) with Web Ontology Language (OWL) to conduct knowledge-based reasoning. Knowledge modules convert data with ros message format into data with Resource Description Framework (RDF) format.

3 Robot Tea House

This section describes how we develop Robot Tea house with PRINTEPS, based on symbolic knowledge representaion including ontologies, business rules, workflow and C-SPARQL for integratitng symbols and stream data.

3.1 Robot Teahouse

We developed a robot teahouse with PRINTEPS. In this section, we describe the overview of the robot teahouse due to limitations of space.

System Overview

The operations of the robot teahouse include receiving customers, guidance to tables, order taking, drink preparation, serving, checkout, and expression of gratitude towards customers.

Figure3 shows the system overview of customer reception at the entrance.

We use Kinect v2 to detect people and get depth sensor data and images. The image sensing modules send the stream data as entrance and table information to ROS environment. Then the entry detection module converts the entrance information to RDF stream and detects the customer entering event using stream reasoning. The leaving seat detection module converts the table information to RDF stream and detects if customers are leaving the seats. Additionally, age and gender estimation module estimates the age range and the gender of customers from images of faces of people. When an event is detected, these modules check the conditions of business rules based on the teahouse ontology. If conditions are satisfied, then robot moves and talks referring to the rule.

ROS Message

We made four ROS messages to represent entrance information. They are Entrance Sensor, Entrance, Customer, and TeahouseObject. Entrance information such as number of people at the entrance, distance between Kinect and each person, and the age and the gender of each person can be described by these ROS messages.

Teahouse Ontology

In the teahouse ontology, classes and properties for location, group, customer, and machine are defined. RDF stream and business rules are described by SWRL based on the teahouse ontology.

Business Rules

In this study, we use the SWRL rules to let the robot move and speak properly in each situation with the use of the teahouse ontology and its instance data. We defined 32 business rules for customer reception at the entrance. They are mainly classified into three types: rules for determining customer's type, rules for determining group, and rules for determining speech content. A customer's types can be estimated by the age range. A group can be estimated by number of customers and each type of the customer. A speech content can be estimated by the number of customers and the group.

Event Detection based on C-SPARQL

In order to detect entry and leaving seat in the robot teahouse, we use C-SPARQL[Barbieri et al., 2010]. C-SPARQL enables event processing when a retrieval language with extended SPARQL is used with RDF stream that employs time-stamped RDF triples.

Figure 4 illustrates the mechanism for entry detection.This query measures at every one-second interval the distance between the face of every person (detected within 3 s) and Kinect. The distances between the faces of the people detected within 3 s (ID: c_1) and Kinect are chronologically shown as c_1d_1, c_1d_2, c_1d_3, and the values of c_1d_1-c_1d_2, c_1d_1-c_1d_3, and c_1d_2-c_1d_3, are computed in order to count the number of values greater than 0.1. The value 0.1 is determined based on the accident error of Kinect's depth sensor value. This measurement is used to avoid erroneously detecting someone who is in front of the teahouse but is not approaching it as a customer. A count exceeding 1 means that someone is approaching the teahouse (Kinect), which is a measurement used to avoid erroneously reacting with a customer who is leaving the teahouse. The average value of the distances between the faces of the people detected within 3 s and Kinect is also calculated, and if the average value is less than the ?distance_to_entrance (3.3m), the customers are recognized as having entered the teahouse, and their IDs, the aforementioned count, and the average values are returned.

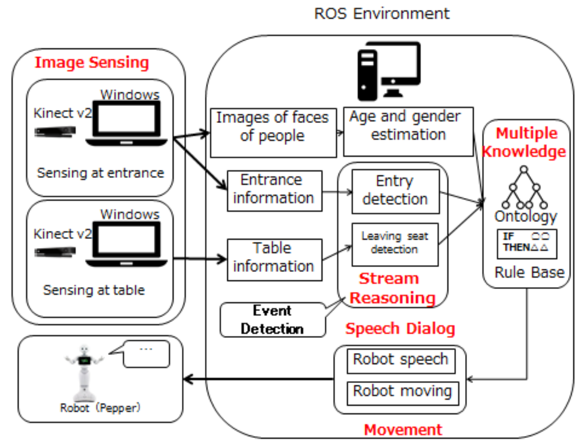

Figure 3 System overview of customer reception at the entrance.

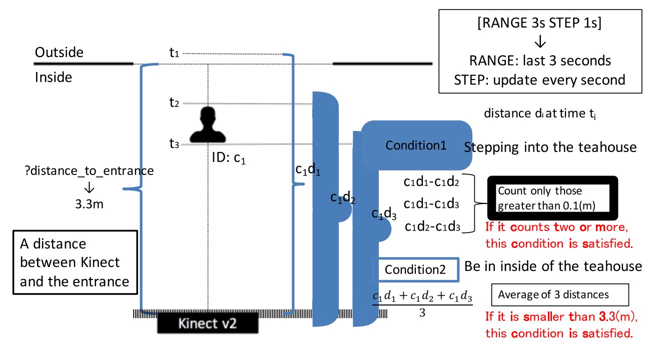

Figure 4 . Mechanism for entry detection.

The value of ?distance_to_entrance is obtained using the SPARQL query for measuring a distance between Kinect and an entrance by predefining, based on the teahouse ontology, the distance between the location in which Kinect is installed and the entrance of the teahouse.

Currently, a distance between a person's face and Kinect is the only information that is used. However, if the sensor can obtain various attribute information from a person in the future, more complex customer detection based on such information (e.g., discerning a customer from a teahouse clerk based on clothing) will be realized.

Workflow

The “entrance sensing" module, using Kinect SDK's person-detection library, continues to output ROS Topics indicating the number of people near the entrance and the distance between people and the Kinect sensor. The ROS Message on “Entrance” output from the “entrance sensing” module is shown in Figure5. The “Entrance.msg” data contains an ID, the number of customers (number_of_customers), and an array of the customer type (has_customers).

4 Experiments

Robot teahouse inckudes seven business processes: (1)greeting customers upon arrival, (2)showing customers to their seat, (3)taking orders, (4)preparing meals, (5)serving, (6)settling bills, and (7)thanking customers upon their departure. We use a single room on campus to construct a mock teahouse space. Figure 5 shows a schematic diagram illustrating the configuration of our robot teahouse system. Pepper works for (1), (2), (3), (7), Jaco2 and HSR for (4). Humans work for (5) and
(6).

4.1 Overview of experiments

We asked customers to experience the seven components of teahouse services while owners of actual teahouses observed as monitors. We then solicited input from the monitor observers regarding their impressions of the interactions and their suggestions for how they could be improved, after which we assessed the feasibility of modifying the robot service to reflect those suggestions. In the experiments discussed here, we had six male and six female monitor observers. We divided the monitors into five groups and conducted one experiment per group, for a total of five experiments. Table 1 shows the composition of the various groups.

4.2 Experimental results

We conducted five experiments, received advice from teahouse owners regarding our operations, and attempted to

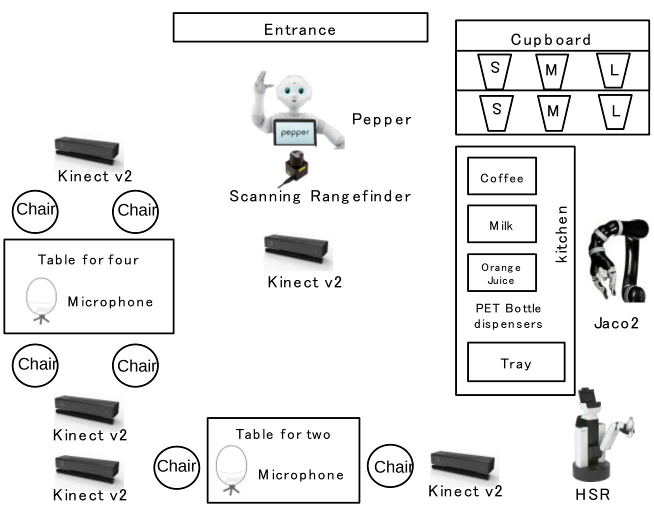

Figure 5 The configuration of our robot teahouse system.

Table 1 The composition of the various groups.

Group	Composition	
	# females	#males
1	2	0
2	1	1
3	2	1
4	1	1
5	0	3

improve performance based on that input. Table 2 lists the suggestions that we received from teahouse owners.

In response to these suggestions, we arrived at a set of proposed modifications. Based on these proposals, we requested modifications to workflow, operating rules, and ontology, and recorded the time required to make these modifications.

In the Category column of Table 2, the phrase “WF” denotes suggestions that may be implemented by using the PRINTEPS workflow editor to add processes or rearrange interconnections and handoffs between workflows. “BR” denotes suggestions that may be implemented by using Drools Workbench to modify conditionals, conclusions, or attributes of existing rules or to create new rules. “Onto” denotes suggestions that may be implemented by modifying the ontology responsible for maintaining teahouse information in PRINTEPS robot teahouses. “SWRL” denotes suggestions that may be implemented by modifying SWRL rules. The symbol “X” denotes suggestions that cannot be implemented with the modules available for use at present but which might become possible to implement by creating new modules or by improving the accuracy or extending the functionality of existing modules. Finally, the symbol “XX” denotes suggestions that cannot be implemented due to limitations on hardware performance, software execution speed, or other factors.

4.3. Discussion

Combining knowledge-based reasoning with image sensing

Using the C-SPARQL event-detection framework to integrate image sensing with knowledge-based reasoning offers three advantages over a direct combination of these paradigms.

First, this approach allows certain detection methods—namely, those based on variations in sensor values due to time evolution—to be expressed in simple form. In this study, we used C-SPARQL to create a triple with a timestamp affixed based on sensor information at each time point. By creating an RDF stream and making search queries of given time widths at given time intervals, we obtain a framework in which searches of multiple data times may be conducted periodically. In contrast, attempting to implement this by directly combining image sensing with knowledge-based reasoning would raise the question of just how such a framework could be realized. In the method that we use here, image sensing and knowledge-based reasoning are completely separated before being combined together, thereby eliminating the difficulty described above and realizing detection methods based on the time variation of sensor values.

A second advantage is the possibility of establishing tie-ins with multiple sensors, which is made possible by separating the image-sensing capability. By separating image sensing from knowledge-based reasoning, it becomes possible to establish interfaces with Kinect and many other sensor types. Moreover, with this configuration, it becomes easy to implement event-detection protocols based on differences in the combinations of various types of sensor data. In our method, we separate image sensing from knowledge-based reasoning and further separate robot action programming. Thus, by implementing a loose-coupling scheme, we allow connections to large numbers of sensors without incurring heavy computational cost.

As a third advantage, the separation of knowledge-based reasoning allows rules to be described in ways that resemble human thought, thereby allowing the exploitation of existing knowledge and expertise. As noted above, when image sensing is connected directly to knowledge-based reasoning, the resulting rules involve complex conditions incorporating physical quantities, which may even require descriptions in the form of embedded code in programs. However, by treating image sensing and knowledge-based reasoning as separate, independent entities, we eliminate the need for

Table2 Suggestions from the teahouse owner.

ID	Suggestion	Category	Time required for modification (HH:mm:ss)
1	Communicate out loud before moving	WF	2:48
2	Make informative bodily gestures before moving	WF	4:57
3	Chef robots should execute some type of performance while idle	WF	8:22
4	When arriving at the table, invite customers to sit down	WF	1:40
5	Teach robots proper timing for responding to customers	WF	2:23
6	Communicate out loud when visiting the table before ordering	WF	1:09
7	Communicate out loud when visiting the table before paying the bill	WF	3:40
8	Identify available seats and the information table and deliver an informative statement together with the welcoming greeting	WF+BR	16:22+5:38
9	Enlarge bodily gestures accompanying welcoming greetings	BR	2:11
10	Ask all customers whether they would like a blanket for their legs	BR	3:19
11	When the table is dirty, inform the customer that the robot is cleaning it	BR	2:08
12	Change the distance between robot and human	Onto	0:31
13	Change the distance between robot and table	Onto	1:00
14	Accelerate the speed of reaction to empty cups	Onto	6:05
15	Reduce the age threshold for providing child seats	SWRL	1:20
16	Slightly decrease the quantity of beverage poured	SWRL	3:43
17	Detect facial expressions and modify the content of spoken communication appropriately	X	
18	Detect facial expressions and modify bodily gestures appropriately	X	
19	Change voice depending on factors such as facial expression and age	X	
20	Change the robot’s direction of rotation	X	
21	Allow customers to manipulate robots in simple ways	X	
22	Display orders on a tablet or display	X	
23	Modify the movement sequence when placing cups on tables	X	
24	Use a stirrer to stir beverages	X	
25	Increase the speed of motion during meal preparation	X	
26	Slightly increase response speed	XX	
27	Increase Pepper’s speed of motion	XX	
28	Move sideways and backwards	XX	
29	Add ice	XX	

direct consideration of image sensing when defining knowledge-based reasoning rules and other elements, which then allows expressions of knowledge to be defined in forms that are similar to the rules used in ordinary operation or to terminology knowledge. This facilitates the utilization of knowledge and expertise accumulated over the course of ordinary operations. However, if it turns out that such elements need to be described at a highly granular level—such as When customers stand up from their seats after settling their bill, thank them and show them to the door—then it will be easy even for operational supervisors to contemplate rules. Since this establishes a connection to ordinary operating experiences, it may also become possible to make modifications that utilize knowledge and expertise. Additionally, because the knowhow (knowledge and expertise) of an operational supervisor is vast and is an extremely important element characterizing operations, the ability to put such knowhow to use may be expected to deliver huge advantages.

Observations regarding robot teahouses

Here we discuss reactions to the robot teahouse experience.

Whereas humans are naturally capable of delivering rapid response times and voice recognition, it is quite difficult for robots to achieve high accuracy in these areas due to execution speed limitations and ambient noise in external environments. This reaction may be attributed to the high expectations customers have of robot hosts, and suggests that improvements such as directing robots to notify customers in advance of their weaknesses; for example, perhaps the robot could say, “I’m a little sluggish in responding, so please bear with me!” before beginning to serve customers. We also learned that the ability of robot chefs to pour beverages properly left a lasting impression, whereas the ability of a human server to pour a beverage leaves no impression at all. Thus, we may think of this as a new service that is made possible precisely because a robot is providing it. At present, we have not chosen a set of metrics for measuring the quality of service at a robot teahouse; instead, we are using the operational wisdom of teahouse owners to improve the services provided. Future work will require defining, from the perspective of service engineering, a set of metrics for assessing quality of service, and then establishing connections to the operational knowledge of teahouse owners in order to improve service at robot teahouses.

5 Related Work

In order to support the development of integrated intelligent applications and/or service robot applications, several tools and platforms have been proposed [Choi et al., 2015, Datta et al., 2012, Lemaignan et al., 2016, Pot et al., 2009, Zander et al., 2016]. However, many works come for developers/researchers and not for end users.

RoboEarth [Tenorth et al., 2013] is a project focused on tasks related to autonomous robots, such as robotics, knowledge processing, and environment sensing for understanding the actual environment. Knowledge processing for Robots (KnowRob)[Tenorth et al., 2013] is one of RoboEarth's subprojects and provides several frameworks for knowledge processing using an ontology, for example, and is closely related to this study. Under KnowRob are structured multiple ontologies, which are used as knowledge expressions of tasks, robots, motions, and environment. KnowRob uses a framework that allows its robots to conduct requested tasks by linking all the knowledge contained in those ontologies. It aims to enable robots to exchange information over the Web.

Regarding the integration of image sensing and knowledge processing, RoboEarth enables a robot to understand not only information it acquires through image sensing, but conceptual information as well, by adding Web Ontology Language (OWL)-based knowledge expressions to the object models that are used for object recognition. Moreover, RoboEarth produces robots of different types to conduct tasks such as delivering a drink to a table by linking OWL-based knowledge expressions to environment maps of multiple types or linking to a combination of necessary programs for certain pre-defined tasks known as action recipes.

Two major differences exist between this study and RoboEarth.

The first is with respect to the targets of image sensing. In RoboEarth's image sensing, these are targets or obstacles necessary for a task for which no time-series changes must be considered. This means that the integration of image sensing and knowledge processing is achieved simply by adding knowledge expressions such as conceptual information to the object models.

The second difference is in the correction frequency of workflow. The chief objective of RoboEarth is to enable robots to conduct specific tasks (e.g., delivering a drink to a table) in various environments based on a common workflow. Therefore, RoboEarth does not require that a domain expert directly and frequently change the workflow.

By contrast, this study requires that a domain expert (e.g., a teahouse owner) change the workflow of robot services frequently based on unique experiences of both robots and customers, the environments in which they operate, and reactions from customers. For that purpose, this study aims to build a platform that allows easy changes based on image sensing results by providing an SOA-based workflow editor that facilitates adjustments to robot services. These changes can also be accomplished by separating business rules from programs or by revising the knowledge processing mechanism.

6 Conclusion

PRINTEPS has two facilities: AI integration and agile process. In particular, the latter makes incremental AI apps development process smartl, accepting domain expert's feedback. Developing robot tea house by PRINTEPS, it takes just less than one hour to revise robot tea house apps by tea house owner's sixteen comments. Comparing with conventional procedual robot service environment, the agility process with PRINTEPS goes better.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR14E3, Japan.

References

[Barbieri et al., 2010] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus. C-SPARQL: A Continuous Query Language for RDF Data Streams. International Journal of Semantic Computing, 4(1):3–25, 2010.

[Choi et al., 2015] Donghee Choi, Jeayeol Ha, Minkyung Jung, Woomin Park, and Hongseong Park. Development of Robot Scenario Script Language and Tool for Non-Expert. Jounal of Automation and Control Engineering, 3(6):498–502, 2015.

[Datta et al., 2012] Chandan Datta, Chandimal Jayawardena, I Han Kuo, and Bruce A MacDonald. RoboStudio - A visual programming environment for rapid authoring and customization of complex services on a personal service robot. IROS, 2012.

[Lemaignan et al., 2016] Séverin Lemaignan, Mathieu Warnier, E Akin Sisbot, Aurélie Clodic, and Rachid Alami. Artificial cognition for social human–robot interaction: An implementation. Artificial Intelligence, July 2016.

[Pot et al., 2009] E Pot, J Monceaux, R Gelin, and B Maisonnier. Choregraphe: a graphical tool for humanoid robot programming. In RO-MAN 2009 - The 18th IEEE In- ternational Symposium on Robot and Human Interactive Communication, pages 46–51. IEEE, 2009.

[Tenorth et al., 2013] M. Tenorth and M. Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. International Journal of Robotics Research, 32(5):566–590, 2013.