

Teaching Sequences to a Social Robot by Voice Interaction

Javi F. Gorostiza and Miguel A. Salichs

Abstract—In this paper a sequence manager system for Robot Teaching is presented. This system allows the user to edit, execute and debug the sequence by means of speech with a multimodal social robot. The main goal of the paper is to make human-robot interaction easier for the non-expert users. To achieve this we are designing a game for children where they play to teach to the robot a sequence of actions and conditions by means of human-robot interaction. A *Sequence Function Chart* (SFC) representation for sequence implementation is proposed. This representation is transparent for the non-expert user, that just uses natural language to interact with the robot. We think that this work will be notable for the development of social robots in our society and to close these robots to common people.

I. INTRODUCTION

Social robots are going to take part of the common life of humans in the developed societies. In 2006 around 3,540,000 service robots were made where only 950,000 were expected [5] Social robots form a new class of computer-based entertainment that is beginning to become commercially practical. We focus on how robots are going to learn new tasks from human, and how to make robots that can educate and entertain the user. The Robot Teaching issue merges concepts of robot learning and human - robot interaction, where the user teaches something to the robot. The major directions in robot learning have been using human learning models for non-supervised learning or programming by demonstration. These *modus operandi* have given very good results: the control architecture proposed in [2] allows the robot to infer the correct goal in a human-robot collaboration scenario; in [9] a system for robot skill acquisition from kinesthetic demonstrations is presented. The implemented system treats low-level motion control of the DOF of a humanoid robot. In other works about programming by demonstration it is proposed a solution of how to imitate (the so called correspondence problem) and what to imitate.

Here we propose a different paradigm where the user teaches or programs complex sequences of actions and conditions in a social robot by speech. A particular goal to achieve is to make a game for children where the child can play to teach sequences to the robot. This system will allow to teach new and complex tasks to the robot in an easy and natural way for non-experts users.

We have chosen a Sequence Function Chart standard [6]

Javi F. Gorostiza is with RoboticsLab, Escuela Politécnica Superior, Universidad Carlos III de Madrid, av. Universidad, 30, Leganes, Spain jgorosti@ing.uc3m.es

Javi F. Gorostiza is with RoboticsLab, Escuela Politécnica Superior, Universidad Carlos III de Madrid, av. Universidad, 30, Leganes, Spain msalichs@ing.uc3m.es

to represent the sequence of actions and conditions in the robot. The sequence can be created, executed and debugged at runtime. Other works have proposed similar representation frameworks. In [11] a high level multi-robot programming representation based on the semantics of Petri nets is implemented. But in this work the expert developer has to program the plans by hand. The plans managed involve a small set of actions and conditions that are very constrained to a model of the environment.

In section II it is made an introduction to SFC standard. In sections III, IV and V the personal robot Maggie and its control architecture are presented as the researching platform of the present paper. In the next section VI it is presented the skill that allows the user to edit a sequence by voice interaction. We present the first experimental results in some cases where the user edits the sequence by means of speech interaction in section VII.

II. SEQUENCE FUNCTION CHART STANDARD AND SEQUENCES

SFC is used here for sequence representation. This graphical programming language is defined by IEC 61131-3 standard for Programmable Logic Controllers (PLC's). Based in the so called GRAFCET (Graphe de commande etape-transition) it is an evolution of a Petri Net that allows the representation of a Sequential System, that is, a system where the outputs do not depend only on the inputs, but on the internal state of the system, too.

A sequence can be defined as a net where each node is an action or a condition. An action is defined as any process or effect of the robot activity. A condition is defined as a functional entity that can take one of two values: true or false, depending on certain variables of the robot or the environment, where we include the user. Then, a sequence is a set of actions, a set of conditions and a set of links between these actions and conditions. SFC is based on two main entities: *steps* and *transitions*. The former are associated with actions and the latter with conditions. A SFC is a bipartite graph, so the connections between nodes of the same type are not allowed, that is, one step cannot be directly linked to another step and *idem* with transitions.

A. Steps and Actions

Symbolically steps are represented as squares with an alphanumeric label. Next to the square the associated actions are defined with a label text. Here we consider that just one action is associated with the step. At a given instant a step may be either active or inactive, what it's represented

as a binary signal associated with the step that it's called *check back signal*. This signal can be used in any condition of a subsequent transition. The set of active steps defines the situations of the system considered.

When a step is active, the associated actions are performed. Graphically this step is marked with a dot, that does not belong to the step symbol and it is only used for explanatory purposes. The initial steps are active at the beginning of the control process. When a step is deactivated, the actions can terminate or maintain its state. In the latter case, the command should be stored, and it will only terminate if it is explicitly reset by a subsequent step.

B. Transitions and Conditions

Symbolically a transition is represented by a dash which is shown in the directed link between the involved step symbols. Every transition has one condition associated. A transition is enable if all the immediately preceding steps, connected to its corresponding transition symbol by directed links are active. Clearing of a transition occurs if it is enable and its associated condition is true. The clearing of a transition simultaneously lead to the activation of all the immediately following connected steps, and the deactivation or the immediately preceding connected steps. The transition condition is a logic proposition that can be true or false.

C. Basic Structures

In every SFC we can find, at most, five elementary patterns that are represented in Fig. 1. These basic structures can appear in different stages.

A single sequence (Fig. 4-a) is made up of a series of steps which will be activated one after another. Each step is followed by only one transition and each transition is enable by one step.

In a sequence selection (Fig. 4-b), an evolution will take place from the initial step to one of the next if the corresponding transition condition is true. In order to select only one sequence, it's necessary that the transition conditions associated with the sequences are exclusive so that they are not true at the same time.

An evolution in a sequence selection convergence will take place from one of the active steps to the final step if the transition condition is true (Fig. 4-c).

Simultaneous sequences (Fig. 4-d) can be activated at the same time from a unique pair step-transition.

In a simultaneous sequence convergence (Fig. 4-e) the evolution will take place only if the steps immediately above the double line are all in the active state and if the transition condition associated with the common transition is true.

III. MAGGIE, SOCIAL ROBOT AND RESEARCH PLATFORM

The present work has been implemented in a social robot, *Maggie*, that is being developed at RoboticsLab in the Universidad Carlos III de Madrid. As it has been completely described in other works like [10] and [3] here we make a brief introduction.

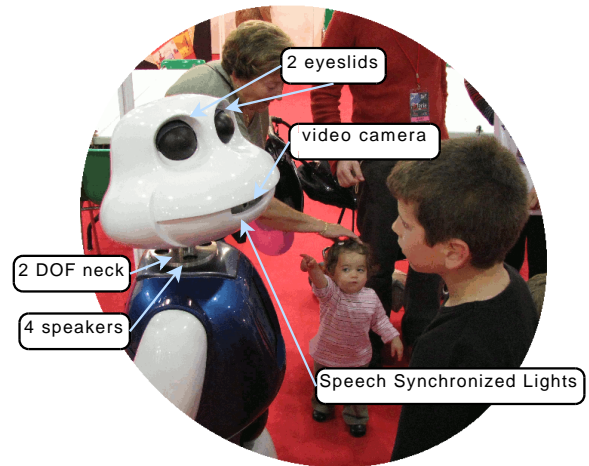


Fig. 2. Maggie interacting with some children

Maggie counts with a friendly external looking. Her functional system is thought for researching in peer-to-peer interaction, edutainment, and robot learning by training or teaching. Maggie is 1.35 meters tall. The base is motorized with two differentially actuated wheels. She has some sensors for navigation: bumpers, infrared optical sensors, ultrasound and laser range finder. The neck has two DOF: left/right and up/down. Maggie has two black eyes with two mobile eyelids. The mouth has speech synchronized lights and a video camera. Two 1-DOF arms are built in the central part of the platform. Among the shell, Maggie counts with several capacitive sensors: in the hands, in the top of the head, in the shoulder, etc. For speech communication, the user counts with a enabled wireless microphone and four speakers. In Fig. 2 we can see an image of the robot interacting with some children.

IV. AUTOMATIC-DELIBERATED ARCHITECTURE

As it has been presented in other works like [1] [3] the AD architecture is based on skills. Each skill works as an independent module that carry out a specific function associated to a capacity of the robot.

There are two layers in the architecture: automatic and deliberated. In the former several skills can be activated at once working in collaboration. In the latter only one skill can take the main control of the system. This level is reserved to "reasoning" skills like a planner, a supervisor or like the **sequence verbal edition skill** that is described later. Sensorimotor and perceptive skills are found in the automatic level.

The distributed skills in the automatic level communicates each other by means of two systems: the Event System and a Short Term Memory System.

The Event System is based on the publisher-subscribe pattern: every skill of the architecture can subscribe to one or

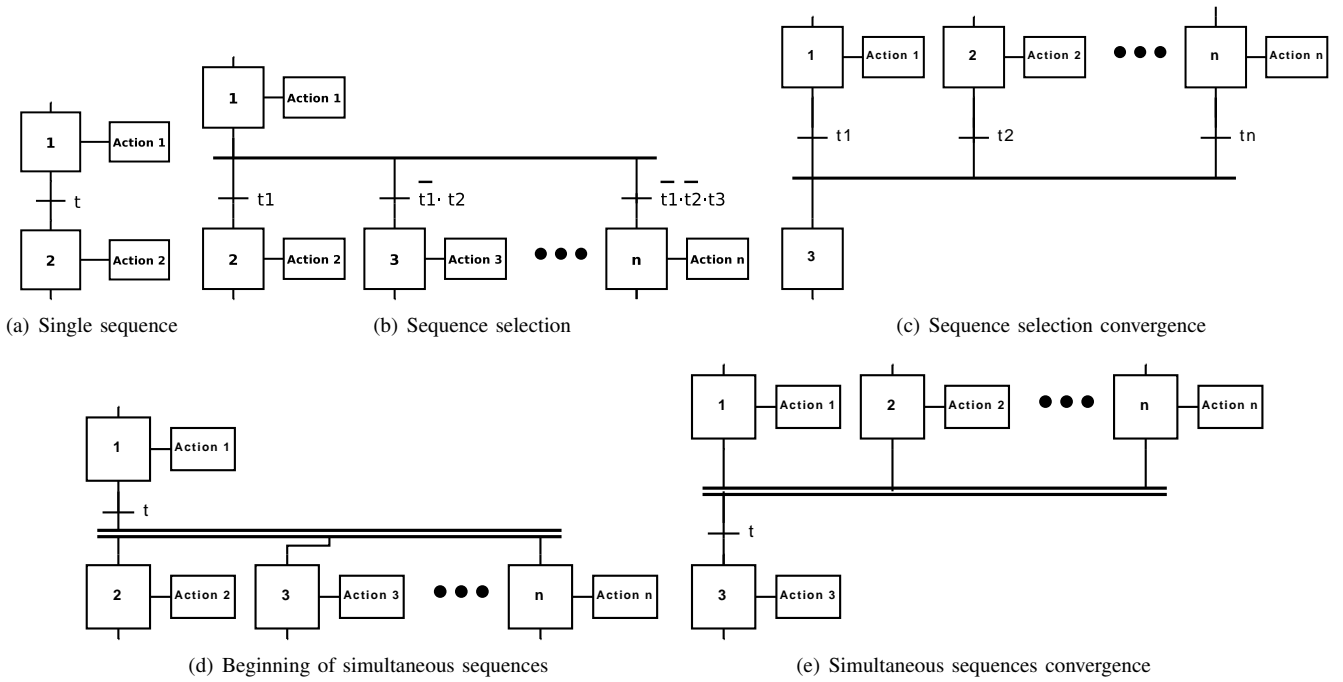


Fig. 1. Basic SFC structures

more events. In the other side, every skill in the system can send an asynchronous event without taking care which subscribed skill is going to catch and manage it. The Event System takes charge of distributing each sent event to the corresponding subscribed skills.

The Short-Term-Memory (STM) works like a shared blackboard where any skill can make data writings and readings. The access to the STM is very fast ($<10\text{ms}$) In this memory it's saved information like sensory data, text-to-speech, text from the automatic speech recognition, etc.

V. SEQUENCE MANAGEMENT SYSTEM

The Sequence Management System takes charge of parsing and executing a sequence. Any skill in the architecture can have one or more instances of the Sequence Management System, so there may be more than one sequence running at once. The sequence is defined in a XML file. Actions and Conditions are represented as following:

Action. It is included in a SFC step. It is associated with two functions: *Activation Function* and *Deactivation Function* The former is executed when the step becomes active, the latter when it is deactivate.

Condition. It is included as a SFC transition. It is associated with a *Transition Function* that is executed always the transition is enabled. This function executes a boolean expression.

The functions associated with steps and transitions are implemented in PYTHON [7] that is an interpreted programming language, and they are embedded in the XML file. These functions can access to the STM and the Event System, and they can communicate with any skill of the architecture. Each instance of a Sequence Management System

parses its XML sequence file so the PYTHON functions are interpreted at runtime.

VI. SEQUENCE EDITION VERBAL SYSTEM

It has been developed a human-robot interaction system where the user can create, edit and debug a sequence of actions and conditions. The edition of the sequence is made by means of user speech.

First of all we have established the limits of the system: what type of users will use it, how it's going to be this usage and what the roles and functions of the system are. Our main purpose is to implement a system for non-expert users, for example, for children to play to a game of teaching sequences to the robot. So Natural Language Processing has been achieved. In Dialogue System Management theory it is defined the so called interaction style depending on who of the two actors is going to have the initiative. So it's distinguished three interaction styles: system initiative, user initiative and mixed initiative [8] In the present research we are not dealing with turn taking and other dynamic processes. Furthermore, what we want to study is how to transform user utterances to sequence elements. So, for simplicity, we have chosen the user initiative interaction style.

The user utters a sentence that is analyzed. The robot gives simple automatic feedback with utterances like *O.K., you want to add a new action, sorry I don't understand, etc.* depending on the successful of asrSkill.

The system allows the user to do two types of actions by means of speech: editing the sequence and executing it. Editing the sequence implies adding and removing any step or transition in any place of the sequence already created, and set the links between them.

First of all we have made a preliminary study about how the user can generate and edit the elements of a sequence by means of speech: what types of utterances (vocabulary and grammars) are used and how they connect to the edition of the sequence elements (nodes and arcs). The robot creates the sequence responding to user speech by means of the so called *Sequence Verbal Edition Skill (sveSkill)*. To manage the sequence net, this skill uses two types of data: nodes and references to the nodes. The former are the nodes of the sequence: steps (actions) and transitions (conditions), the latter represents the focus of attention in the dialogue between the user and the robot. In natural language, it is usual that the participants assume references to concepts that have been just referenced before, so sveSkill saves actions and transitions in focus, so they can be used in consecutive user utterances. This idea is based on other works about attention and the structure of discourse like [4]. Therefore the implemented system allows the user to add/remove any SFC step or transition in any part of the sequence net using some specific natural language utterances.

The set of possible actions and conditions correspond to a set of activate, deactivate and transition functions that are saved as data in the system memory. The interface between the user and the set of actions and conditions are *grammars*. Grammars are used by the asrSkill to link user utterance with the functionality of the robot. They work as interfaces for the user to make reference to the actions and conditions of the system. Grammars are saved as data in the system memory, and can be changed at runtime. So, to add a new action or condition in the system it is enough to add the corresponding activate, deactivate and transition functions and a reference in an ASR-grammar, and it is not necessary to make any change in the system implementation.

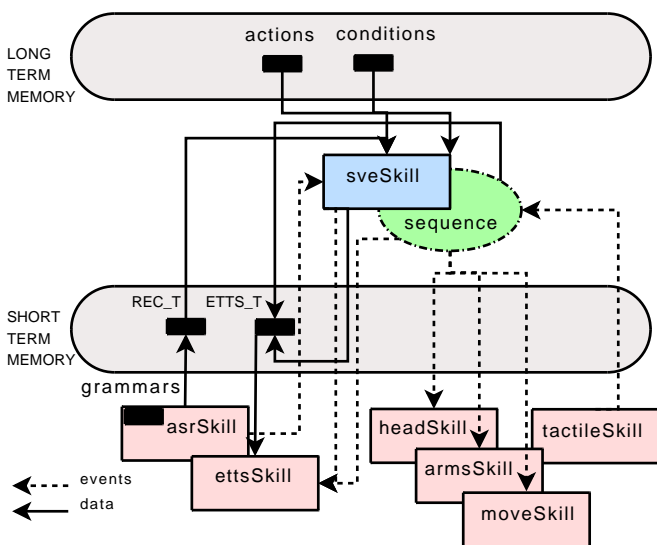


Fig. 3. Architecture of an example of use of the

VII. EXPERIMENTAL RESULTS

A. Example with some Automatic Skills

In this section some of the automatic skills implemented in Maggie are explained. They implement an example of use of the Sequence Edition Verbal System. In Fig. 3 it is represented a scheme of the different involved skills and their connections inside the AD architecture.

Two voice skills have been implemented: Automatic Speech Recognition Skill (asrSkill) and Emotional Text-to-Speech Synthesis Skill (ettsSkill). The asrSkill is configurable for multiple languages. It doesn't require a training process so it's speaker independent. It can dynamically load and unload grammars in EBNF format. Barge-in feature is also available. After the user utterance is processed an event with an associated parameter is emitted to the Event System. The parameter represents the grammar identification corresponding to the grammar number that best fits with user utterance, or a special value in case of misunderstanding. If the recognition succeeds the recognized words with confidence values are written in the STM. The ettsSkill synthesizes speech from text. It controls some prosodic speech parameters that allows certain emotional speech expression and three intentional modals: declarative, exclamative and interrogative. The synthesized text is read from the STM. Movement and rotation is made by means of baseSkill that connects the robot-base engines to the rest of the control architecture. The arms movements are made by means of armsSkill; neck and eyes movements are made by means of headSkill. As mentioned above Maggie counts with several capacitive sensors around her shell. These two-state sensors are activated when, for example, a human hand approaches close to the sensor antenna. When this happens, an event with sensor state information is emitted.

B. Case Studies

The user can add an action to the sequence just saying the specific command. For example: *Raise the left arm* or *Turn right*. When an utterance like that is detected by the asrSkill, the sveSkill adds a step to the net with its corresponding action. Transition addition is made analogously from user utterances like *If I touch your head...*. The system is able to know if the added step is an initial step to be activated when the sequence begins to execute, or if the new added node has to be connected to another existing node.

In Fig. 4 we can see two simple examples of node addition. In the first example an explicit condition is added after the first action. In the second example the condition is implicit: the system interprets the utterance *then...* like the user wants to add another action to be executed when the previous has finished.

In Fig 5 it is shown an example where the user adds three actions that run concurrently after a single condition. In this example we can see that the user can do an implicit or an explicit reference to an added action. In the utterance *At the same time...* the system is able to complete the information and understand that the users is referencing the action *Raise*

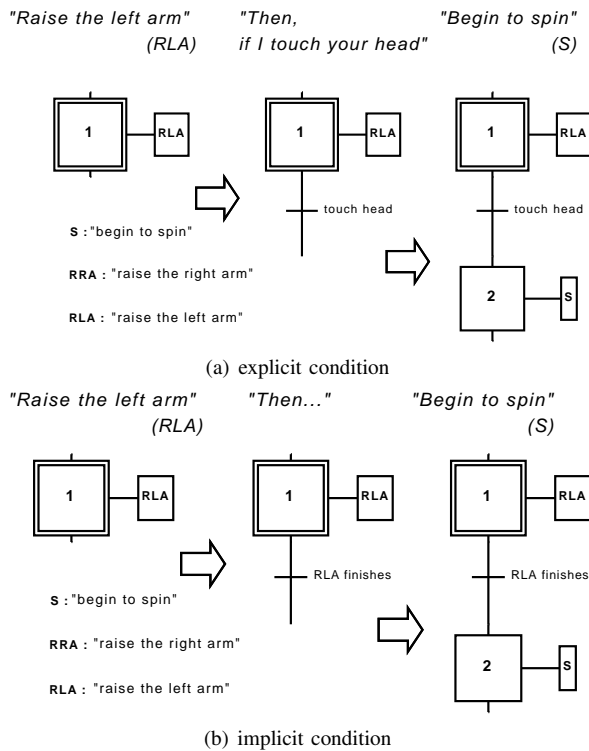


Fig. 4. node addition vs. user utterances

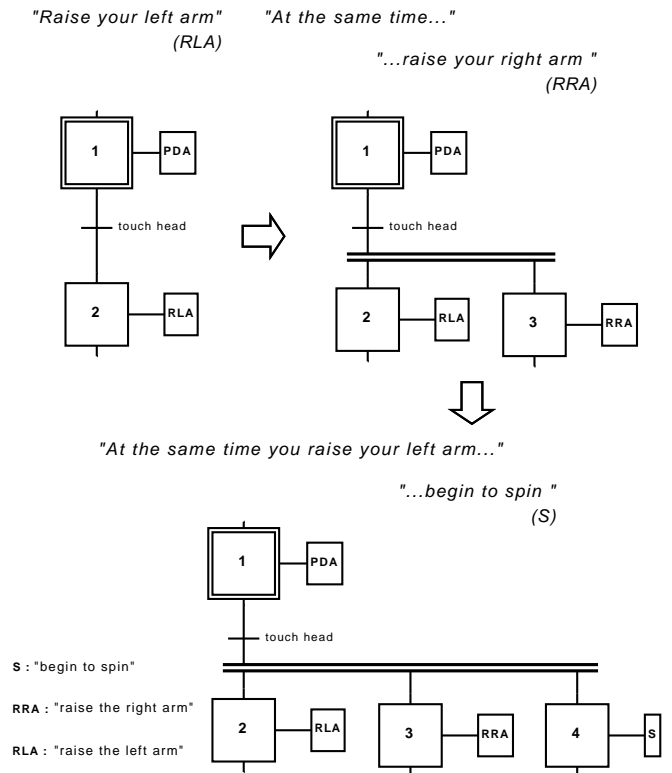


Fig. 5. node addition vs. user utterances

your left arm because it is the last mentioned one. However, in the next referencing utterance *At the same time you raise your left arm...* an explicit reference to the action is made. In both cases the system adds the action in the same *level of hierarchy*, that is, connecting the new action to the same condition of the referenced action that activates it.

When the sequence is created, the user can command by voice to execute it to probe if it is what he/she intended to do. The sequence can be stopped and modified at any time by means of voice interaction.

VIII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We have designed and implemented a human-robot interaction system where a non-expert user can edit, execute and debug a sequence to a multimodal social robot by means of voice interaction. We have probed that the *Sequence Function Chart* standard is a powerful language for sequence representation, as it allows to depict a bipartite net of actions and conditions where multiple patterns can be found. The sequence is implemented as a XML data, that can be easily changed at runtime, without making any change in the system.

B. Future Works

Actually the robot counts with an initial set of actions and conditions that the user utilizes to create his/her own sequences. As they are represented as interpreted functions in the memory of the system, these sets can be change at runtime, so more actions and/or conditions can be added

without making any change in the global system, e. g., a new created sequence could be added in the actions set.

Naturally, to make the new added action or condition accessible for the user, a corresponding grammar has also to be added in the memory of the system, so the *asrSkill* could detect when the user is mentioning the new action or condition, and add or remove it in any part of the sequence in focus. This is also supported by the architecture implementation. We conclude that the way that actions and conditions are included in the system allows scaling these sets without making any changes on the implemented system. At the moment we also are developing more sophisticated Dialogue Management System, including error recovery modules, mixed initiative dialogs, and multimodal expression feedback (not just TTS acknowledgment). Multimodality perception can also improve the robot teaching process as non-speech modals give extra information in a quick and precise way.

IX. ACKNOWLEDGMENTS

The authors gratefully acknowledge the funds provided by the Spanish Ministry of Education and Science (MEC) through the projects named "Personal Robotic Assistant" (PRA) and "Peer to Peer Robot-Human Interaction" (R2H), of MEC (Ministry of Science and Education)

REFERENCES

- [1] R. Barber and M. Salichs. A new human based architecture for intelligent autonomous robots. In *The Fourth IFAC Symposium on Intelligent Autonomous Vehicles*, pages 85–89, 2001.
- [2] C. Breazeal, A. Brooks, D. Chilongo, J. Gray, A. Hoffman, C. K. H. Lee, J. Lieberman, and A. Lockered. Working collaboratively with humanoid robots. *ACM Computers in Entertainment*, 2(3), July 2004.
- [3] J. F. Gorostiza, M. A. Salichs, R. Barber, A. Khamis, M. Malfaz, R. Pacheco, R. Rivas, A. Corrales, E. Delgado, and D. García. Multimodal human-robot interaction framework for a personal robot. In *RO-MAN 06: The 15th IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, U.K., September 2006. IEEE.
- [4] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.
- [5] E. Guizzo. 10 stats you should know about robots but never bothered googling up, 2008.
- [6] I.E.C. *Preparation of Control Chart for Control Systems: Sequential Function Chart (SFC) Standard*. IEC 61131-3, 1993.
- [7] A. Martelli. *Python in a Nutshell (In a Nutshell (O'Reilly))*. O'Reilly Media, Inc., 2006.
- [8] M. F. McTear. *Spoken Dialogue Technology, Toward the Conversational User Interface*. Springer, 2004.
- [9] S. C. Micha Hersch, Florent Guenter and A. Billard. Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Transactions on Robotics*, 24(6):1463–1467, 2008.
- [10] M. A. Salichs, R. Barber, A. Khamis, M. Malfaz, J. Gorostiza, R. Pacheco, R. Rivas, A. Corrales, E. Delgado, and D. García. Maggie: A robotic platform for human-robot social interaction. In *Submitted to IEEE International Conference on Robotics, Automation and Mechatronics (RAM 2006)*, Bangkok, Thailand, 2006. IEEE.
- [11] V. A. Ziparo, L. Iocchi, D. Nardi, P. F. Palamara, and H. Costelha. Petri net plans: a formal model for representation and execution of multi-robot plans. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 79–86, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.