Mälardalen University Press Licentiate Theses
No. 149

HUMAN ROBOT INTERACTION SOLUTIONS FOR INTUITIVE INDUSTRIAL ROBOT PROGRAMMING

Batu Akan

2012

MÄLARDALEN UNIVERSITY
SWEDEN
School of Innovation, Design and Engineering

ISBN 978-91-7485-060-4
ISSN 1651-9256
Printed by Mälardalen University, Västerås, Sweden

Abstract

Over the past few decades the use of industrial robots has increased the efficiency as well as competitiveness of many companies. Despite this fact, in many cases, robot automation investments are considered to be technically challenging. In addition, for most small and medium sized enterprises (SME) this process is associated with high costs. Due to their continuously changing product lines, reprogramming costs are likely to exceed installation costs by a large margin. Furthermore, traditional programming methods for industrial robots are too complex for an inexperienced robot programmer, thus assistance from a robot programming expert is often needed. We hypothesize that in order to make industrial robots more common within the SME sector, the robots should be reprogrammable by technicians or manufacturing engineers rather than robot programming experts. In this thesis we propose a high-level natural language framework for interacting with industrial robots through an instructional programming environment for the user. The ultimate goal of this thesis is to bring robot programming to a stage where it is as easy as working together with a colleague.

In this thesis we mainly address two issues. The first issue is to make interaction with a robot easier and more natural through a multimodal framework. The proposed language architecture makes it possible to manipulate, pick or place objects in a scene through high level commands. Interaction with simple voice commands and gestures enables the manufacturing engineer to focus on the task itself, rather than programming issues of the robot. This approach shifts the focus of industrial robot programming from the coordinate based programming paradigm, which currently dominates the field, to an object based programming scheme.

The second issue addressed is a general framework for implementing multimodal interfaces. There have been numerous efforts to implement multimodal interfaces for computers and robots, but there is no general standard framework for developing them. The general framework proposed in this thesis is designed to perform natural language understanding, multimodal integration and semantic analysis with an incremental pipeline and includes a novel multimodal grammar language, which is used for multimodal presentation and semantic meaning generation.

Annem için

Acknowledgments

My journey in Sweden has been a long one, but I have known my co-supervisor Baran Çürüklü for even longer. We have discussed about many things, from cameras, guitars, whiskey, Japanese kitchen knives, to why the pipes of the buildings in Sweden are inside rather than outside, but most importantly lots and lots of research. Lots of questions and ideas going around the room in heated discussions, which I enjoyed very much (most of the time). I could not have written this thesis in fact I wouldn't even be here writing these lines without the support of Baran.

Many thanks go to my supervisors Lars Asplund and Baran Çürüklü for teaching me a lot of new stuff, for guidance and support, for all the fruitful discussions, and for the company during the conference trips.

Many thanks go to, Fredrik Ekstrand, Carl Ahlberg, Jörgen Lidholm, Leo Hatvani and Stefan (Bob) Bygde for all the funny stuff, the humor, the support and for sharing the office space with me where working is both fruitful and fun. I owe many thanks to Afshin Ameri for helping me as co-author and as friend. I would like to thank many more people at this department, Adnan and Aida Čaušević, Aneta Vulgarakis, Antonio Ciccheti, Cristina Seceleanu, Dag Nyström, Farhang Nemati, Giacomo Spampinato, Hüseyin Aysan, Jagadish Suryadeva, Josip Maraš, Juraj Feljan, Kathrin Dannmann, Luka Lednički, Mikael Åsberg, Moris Benham, Nikola Petrovič, Radu Dobrin, Séverine Sentilles, Thomas Nolte, Tiberiu Seceleanu, and Yue Lu for all the fun coffee breaks, lunches, parties, whispering sessions and the crazy ideas such as having meta printers that could print printers for printing anything.

I dont know where I would be if it was not for Ingemar Reyier, Johan Ernlund and Anders Thunell. Thank you for helping me with many technical and theoretical challenges that I have had.

Along the way I picked up lots of new and precious friends both in and outside the university environment and without whom I believe I could not have continued further. Thank you Burak Tunca, Cihan Kökler, Cem Hizli, Clover Giles, Katharina Steiner and Fanny Ängvall for being my friends along this path.

I would like to express my gratitude to my parents Nimet Ersoy and Mehmet Akan as well as to my sister Banu Akan for their unconditional love and support through out my life.

This project is funded by Robotdalen, VINNOVA, Sparbanksstiftelsen Nya, EU European Regional Development Fund.

Thank you all!!

Batu Akan
Västerås, March 28, 2012

List of Publications

Papers included in the thesis [1]

Paper A *Interacting with Industrial Robots Through a Multimodal Language and Sensory Systems*, Batu Akan, Baran Çürüklü, Lars Asplund, 39th International Symposium on Robotics, p 66-69, Seoul, Korea, October, 2008

Paper B *Object Selection Using a Spatial Language for Flexible Assembly*, Batu Akan, Baran Çürüklü, Giacomo Spampinato, Lars Asplund, In Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'09), p 1-6, Mallorca, Spain, September, 2009.

Paper C *A General Framework for Incremental Processing of Multimodal Inputs*, Afshin Ameri E., Batu Akan, Baran Çürüklü, Lars Asplund, In Proceedings of the 13th International Conference on Multimodal Interaction (ICMI'11), p 225-228, Alicante, Spain, November, 2011.

Paper D *Intuitive Industrial Robot Programming Through Incremental Multimodal Language and Augmented Reality*, Batu Akan, Afshin Ameri E., Baran Çürüklü, Lars Asplund, In proceedings of the IEEE International Conference on Robotics and Automation (ICRA'11), p 3934-3939, Shanghai, China, May, 2011.

[1] The included articles are reformatted to comply with the licentiate thesis specifications

Other relevant publications

Conferences and Workshops

- *Augmented Reality Meets Industry: Interactive Robot Programming*, Afshin Ameri E., Batu Akan, Baran Çürüklü, SIGRAD, Svenska Lokalavdelningen av Eurographics, p 55-58 Västerås, Sweden, 2010

- *Incremental Multimodal Interface for Human-Robot Interaction*, Afshin Ameri E., Batu Akan, Baran Çürüklü, 15th IEEE International Conference on Emerging Technologies and Factory Automation, p 1-4, Bilbao, Spain, September, 2010

- *Towards Industrial Robots with Human Like Moral Responsibilities*, Baran Çürüklü, Gordana Dodig-Crnkovic, Batu Akan, 5th ACM/IEEE International Conference on Human-Robot Interaction, p 85-86, Osaka, Japan, March, 2010

- *Towards Robust Human Robot Collaboration in Industrial Environments*, Batu Akan, Baran Çürüklü, Giacomo Spampinato, Lars Asplund, 5th ACM/IEEE International Conference on Human-Robot Interaction, p 71-72 , Osaka, Japan, March, 2010

- *Object Selection Using a Spatial Language for Flexible Assembly*, Batu Akan, Baran Çürüklü, Giacomo Spampinato, Lars Asplund, SWAR, p 1-2, Västerås, September, 2009

- *Gesture Recognition Using Evolution Strategy Neural Network*, Johan Hägg, Batu Akan, Baran Çürüklü, Lars Asplund, ETFA 2008, p 245-248, IEEE, Hamburg, Germany, September, 2008

Contents

I

Thesis

Chapter 1

Introduction

Robots have become more powerful and intelligent over the last decades. Companies producing mass market products such as car industries have been using industrial robots for machine tending, joining and welding metal sheets for several decades now. Thus, in many cases an investment in industrial robots is seen as a vital step that will strengthen a company's position in the market because it will increase the production rate. However, in small and medium enterprises (SME's) robots are not commonly found. Even though the hardware cost of industrial robots has decreased, the integration and programming costs for robots make them unfavorable among SME's. No matter how simple the production process might be, to integrate the robot, one has to rely on a robot programming expert. Either the company will have to setup a software department responsible for programming the robots or outsource this need. In both cases the financial investments do not pay up [1].

Also an industrial robot must be placed in a cell that will occupy valuable workspace and maybe operate only a couple of hours a day. It is, thus, very hard to motivate an SME, which is constantly under pressure, to carry out a risky investment in robot automation. Obviously, these issues result in challenges with regard to high costs, limited flexibility, and reduced productivity.

In order to make industrial robots more favorable in the SME sector, the issues of flexibility have to be resolved. Typically for those SMEs, that have frequently changing applications, it is quite expensive to afford a professional programmer or technician, therefore an human robot interaction solution is demanded. Using a high-level language, which hides the low-level programming from the user, will enable a technician or a manufacturing engineer who has knowledge about the manufacturing process to easily program the robot and to let the robot switch between previously learned tasks.

In this thesis we propose a novel context dependent multi modal language which is backed up by an augmented reality interface that enables the operator to interact with an industrial robot. The proposed language architecture makes it possible to give high-level commands to manipulate, pick or place the objects in the scene. Such a language shifts the focus of industrial robot programming from coordinate based programming paradigm to object based programming scheme.

1.1 Outline of thesis

The remainder of this thesis consists of two main parts. The first part contains five chapters: Chapter 2 gives a brief survey of Human Robot Interaction (HRI) and related basic concepts. Chapter 3 provides an overview of robot programming systems in general. Chapter 4 presents the results achieved during the work with this thesis, and finally, Chapter 5 concludes and summarizes the thesis, and gives directions to possible future work. The second part of the thesis is a collection of four peer-reviewed conference and workshop papers.

Chapter 2

Human-Robot Interaction (HRI)

Robots are artificial agents with capacities of perception and action in the physical world. As robotic technology develops and the robots start moving out of the research laboratories in to the real world, the interaction between robots and humans becomes more important. Human Robot Interaction (HRI) is the field of study that tries to understand, design and evaluate robotic systems for use by or with humans [2].

Communication of any sort between humans and robots can be regarded as interaction. Communication can be of many different forms. However, the distance between the human and the robot alters the nature of communication. Communication, and thus interaction, can be divided into the following two categories: proximate interaction and remote interaction. In proximate interaction, the user and the robot are in the same environment, for example, industrial robots and the user are in the same cell during the programming phase. In remote interaction, the user and the robot can be spatially and temporally separated from each other. For example Mars rovers are both temporally and spatially separated from their users. This division helps to distinguish between applications that require mobility, physical manipulation or social interaction. For example tele-operation and tele-manipulations use remote interaction to control mobile remote robot and manipulate objects that are not in the immediate surrounding of the user, whereas proximate interaction, lets say with a mobile service robot, requires social interaction [2].

In social interactions, the robots and the humans interact as peers or companions, however the important factor is that social interaction often requires close proximity.

While the distance between the robot and the user alters the nature of communication, it doesn't define the level or shape of the interaction. Like many complex systems, a designer, designs the nature of interaction. The designer, attempts to understand and shape this interaction process in the hope of making it more beneficial for the user. From the designers point of view, the following five attributes can be altered to affect the interaction process:

- Level and behavior of autonomy
- Nature of the information exchange
- Structure of the team
- Adaptation, learning and training of users and the robot
- Shape of the task

The remainder of this chapter will discuss these attributes in detail.

2.1 Levels of Autonomy

Robots that can perform the desired tasks in an unstructured environment without human intervention are autonomous. From an operational point of view, the amount of time during which the robot can be left without supervision is an important characteristic of autonomy. A robot with high autonomy can be left alone for longer periods of time whereas a robot with lower autonomy needs continuous supervision and user control. Autonomy, however, is not the highest achievable goal in the field of HRI, but only a means to support productive interaction. Therefore in a human centered applications the notion of *levels of autonomy (LOA)* gains more importance. Even though there are many scales for LOA the following one proposed by Sheridan [3] is the most cited one [2]:

1. Computer offers no assistance; human does it all.
2. Computer offers a complete set of action alternatives.
3. Computer narrows the selection down to a few choices.

4. Computer suggests a single action.
5. Computer executes that action if the user approves.
6. Computer allows the human limited time to veto before automatic execution.
7. Computer executes automatically then necessarily informs the human.
8. Computer informs human after automatic execution only if human asks.
9. Computer informs human after automatic execution only if it decides too.
10. Computer decides everything and acts autonomously, ignoring the human.

However, such scales may not always be applicable to the whole problem domain but are more beneficial when applied to the subtasks in the domain.

The scale proposed by Sheridan helps to determine how autonomous a robot is under certain circumstances, but it does not help to evaluate the level of interaction between the user and the robot from an HRI point of view. For example a service robot should exhibit different levels of autonomy during the programming phase and the execution phase. Figure 2.1 presents a different perspective of autonomy regarding the level of interaction between the robot and the user. It should be noted that one end of the scale doesn't mean less autonomy and the other end more autonomy. For example, on the direct control side of the scale, the issue arises to make a user interface that minimizes the operator's cognitive load. At the other end of the scale, the problem is how to create robots with the appropriate cognitive skills to interact naturally and efficiently with a human to achieve peer-to-peer collaboration [2]. Peer-to-peer collaboration requires not only full autonomy at sub-level tasks at times but also social skills when interacting with humans; therefore it is often considered more difficult to achieve than full autonomy alone.

2.2 Nature of Information Exchange

Autonomy is only one aspect that governs the interaction between the human and the robot. The second component defines how the information is exchanged. Input modality defines the nature of the interaction between the robot and the user. Different modalities carry different types of information.

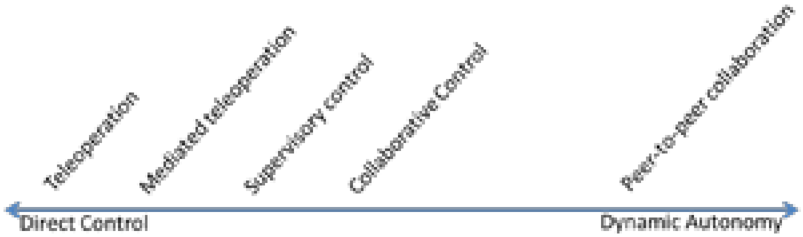

Figure 2.1: Levels of autonomy with emphasis on human interaction.

While interacting with computers or robots often three of the five senses we use are utilized: audio, visual and touch. However, the same message can be carried over two different channels that address two different senses. For example, text and speech may carry the same information but appeal to two different senses. Both represent verbal information exchange, but speech carries additional channels such as tonality so the information exchange really has two dimensions: verbal and non-verbal. Verbal communication could be better suited for passing commands to the robot and non-verbal communication through gestures is more suitable for conveying spatial information. However, combining these two modalities would yield a more complete and richer communication between the robot and the user.

2.2.1 Verbal

Speech is an important modality for exchanging information between the robot and the user. The user can give speech commands to the robot to make it interact with the objects in the scene. These commands can be like: “Pick up the blue object” or “Put it next to the green object”. It is also possible to adjust the settings for the task. The user may command the robot to go “faster”, or “slower”, etc. These commands will enable the user to fine-tune the tasks and the skills. Also, any skills or tasks that have been taught to the robot, can be executed through these speech commands.

2.2.2 Non-Verbal Communication

A gesture is a form of non-verbal communication where visible bodily actions communicate particular messages. Gestures include movement of the hands, face, or other parts of the body. Gestures differ from physical non-verbal communication that does not communicate specific messages. Gestures can be static or dynamic meaning that, the gesture can be a certain pose of the hand or the body or a movement in certain predetermined patterns. For example, a pointing gesture is a static gesture while drawing a circle figure is a dynamic gesture. Gestures can be used to program or control robots [4]. Voyles and Khosia integrated a gesture based set of commands into a programming by demonstration framework [5]. Strobel *et al.* use static gestures to direct the attention of a robot to a specific part of the scene [6].

2.2.3 Multimodal Approach

Speech, facial gestures, body gestures, images, etc. are different information channels that humans use in their everyday interaction, and mostly they use more than one a time. On the other hand, humans see robots as objects with human-like qualities [7]. Consequently, a human-like interaction interface for robots will lead to richer communication between humans and robots. In-person communication between humans is a multimodal and incremental process [8]. Multimodality is believed to produce more reliable semantic meanings out of error-prone input modes, since the inputs contain complementary information which can be used to remove vague data [9].

Although giving instructions to the robot using speech is an intuitive modality for many users, it is not always convenient or sufficient for passing lower-level details to the robot about the skill to be learned. These low-level details can concern the spatial relations between objects in the working environment, or the target orientation and position of the objects. Humans often make use of hand gestures and body postures when information conveyed through speech modality is ambiguous or not detailed enough to describe the task. Therefore it is important to design the language that the user employs to interact with the robot in such a way that multimodal input is accepted.

Since the introduction of the “Media Room” in Richard A. Bolts paper [10], many other systems have been implemented which are based on multimodal interaction with the user. Researchers have employed different methods in implementation of such systems [11, 12, 13, 14]. All multimodal systems are common in the sense that they receive inputs from different modalities and combine the information to build a joint semantic meaning from these inputs.

Finite-state multimodal parsing has been studied by Johnston and Bangalore and they present a method to apply finite-state transducers for parsing inputs [13]. Unification-based approaches are also studied by Johnston [15].

Fugen and Holzapfels research on tight coupling of speech recognition and dialog management shows that the performance of a system can be improved if it is coupled with a dialog manager [16].

A good study on incremental natural language processing and its integration with a vision system can be found here [8] and also in [17]. Incremental parsers are also studied for translation purposes [12]. Schlangen and Skantz have proposed an abstract model for incremental dialog processing in [14].

A multimodal communication scheme is very useful for robot programming because combining two or more modalities can even provide improved robustness [18]. McGuire *et al.* make use of speech and static gestures in order to draw the robot's attention to an object to be grasped [19, 20].

2.3 Structure of the Team

It is often the case that interaction is not limited to one user and one robot. There can be cases where a person needs to command and interact with multiple robots, or multiple users with different roles interact with a single robot, or multiple users interact with multiple robots. Robots used in search and rescue operations are often operated by two humans, with special roles in the team [21] is an example of many to one interaction. On the other hand many Unmanned/Uninhabited Air Vehicles (UAVs) can be controlled simultaneously by a single operator [22].

Designing the structure of the team is another aspect of HRI. There are several questions that arise in this respect: Who has the authority to make certain decisions: the human, the software interface or the robot? Who has the authority to instruct the robot and at which level? How are conflicts solved? How are the roles of the robot and the user defined. The question of what is the role of the human has recently gained importance [23]. Often robots may need to interact with humans who are bystanders with no training at all. For example, a health-assistant robot must help patients and interact with visitors.

2.4 Adaptation, Learning and Training

How to give robots the ability to adapt and learn has been widely researched in academia. However, training of users has received relatively little attention, due to the fact that HRI researchers often want to create robot systems, that can be used with little to no training on the users side. Even the HRI researchers might receive training to create better interactive systems. This section addresses the issues regarding the training of operators, designers, as well as robots. The training is often given in the hope of understanding and improving the user interface, interpreting video feedback, controlling the robot, coordinating with other team members and staying safe while operating the robot in a hostile environment [24].

2.4.1 Efforts to Train Users

Even though one of the goals of good HRI is to minimize the training of the users, it is always necessary required to give careful training to the users in cases where the operator workload or risk is too high. Examples of such cases are military and law enforcement applications, space applications, and search and rescue operations. On the other hand, robots that interact with humans socially are often designed to change, educate or train their users, especially in longterm interactions [25].

2.4.2 Training Designers

The training of designers has received little attention in the HRI literature; however, it is important that they do receive training in the procedures and practices in the fields they seek to help. There are workshops and tutorials for search and rescue robotics [26] as well as tutorials on metrics and experiment design for robot applications [27].

2.4.3 Training Robots

It is often the case that robots need to learn and adapt to the environment or to the user once they leave the factory or the laboratory where they are preprogrammed regarding certain skills and behaviors. However, a well-designed robot, that is beneficial to the user, continues to learn and adapt, by improving both perceptual and reasoning capabilities through interaction. Approaches to robot learning include teaching/programming by demonstration

(PbD), as well as task and skill learning. Methods for programming robots are discussed in more detail in Chapter 3.

2.5 Task Shaping

As new technology is introduced to our lives, the way we do certain things changes. Similarly introduction of new robotic technologies allow a human to do things that they were not capable of doing before, or it eases the physical or cognitive workload by making the task easier or more pleasant to do for the human. This means that introducing new technology fundamentally changes the way humans do the task. Task shaping is the term that emphasizes the importance of considering how the task should be done and will be done as new technology is introduced [2].

Chapter 3

Robot Programming Systems

In this chapter we give a brief overview of how robots are programmed. This chapter follows a similar pattern for dividing the field of robot programming as Biggs na d Macdonald [28]. The field of robot programming is divided in to two: manual programming and automatic programming (Fig 3.1). In manual programming the code is hand created, and this is done through either text-based programming and graphical programming. Automatic programming can be divided in to three sub groups: learning systems, programming by demonstration and instructive systems. In automatic mode the robot code is automatically generated and the user has little or no direct control over the code.

3.1 Manual Programming Systems

Manual programming systems require the user to create the program by hand often without the actual robot. Once the program is finished, it is loaded into the robot and tested. Manual programming systems are offline methods for programming robots, because the code is created either without using the robot, or with the robot disconnected from the programming environment. However when there are no safety concerns, for example while programming toy robots, it might be possible to control the robot online through an interpreted language where line-by-line execution is possible while creating the code.

Manual programming systems can be divided into two groups: text-based systems and graphical programming environments. Graphical programming

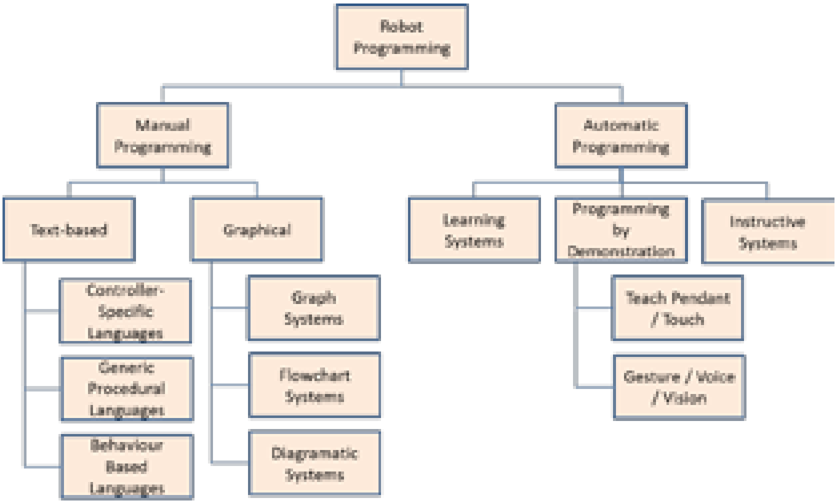

Figure 3.1: Categories of robot programming systems

environments are not considered automatic code generation, because the user must create the code by hand before running it on the robot. Besides there is a one-to-one correspondence with the icons and the generated program statements.

3.1.1 Text-Based Systems

In text-based programming systems, the conventional programming approach is used, and it is the most common way to program industrial robots together with programming by demonstration (PbD). Text-based systems can be grouped depending on the type of language used, by means of the programming done by the user.

Controller-Specific Languages

Controller-specific languages are the most common method to program industrial robots. Ever since the invention of industrial robots and robot controllers there has been a machine language and usually a programming language to go with it that can be used to create programs for that robot. These languages

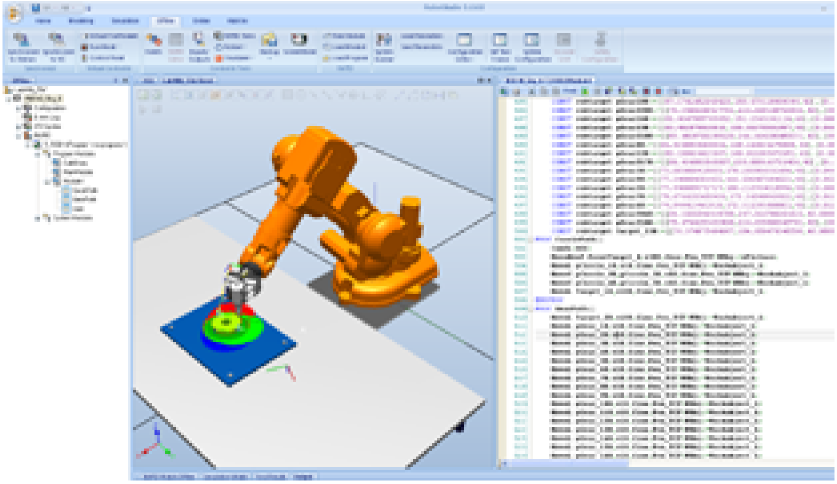

Figure 3.2: A screenshot of the ABB Robot studio.

often consists of very simple commands for controlling the robot, input output (IO), and program flow. ABB's RAPID programming language [29] is a good example of controller specific languages (Fig. 3.2). However KUKA and other manufacturers have their own specific languages targeted for their robots.

In fact, it can be guessed that, there are as many languages as there are manufacturers. The major disadvantage of controller specific languages is that there are no international standard between different manufacturers. If a company owns robots from several manufacturers, the programmers need to be trained for each type of robot or the company will need to outsource robot programming.

Generic Procedural Languages

Generic procedural languages provide an alternative to controller specific languages. Generic programming languages for robots extends standard high level procedural languages such as C++ or Java in a way to provide functionality for the target robot platform. Such an approach is beneficial in research environments, where generic languages are extended to meet the needs of the research project. The extended generic language can be used for system programming or application level programming.

The abstraction provided by the generic language, which consists of a set of classes and functions, can facilitate the programming. Abstraction provides an easy way to control the robot, while hiding away the low-level functionality, such as handling IO's or raw sensor data. For example, with the help of the functions provided the robot can easily be moved to a certain position. Another example is collision-free paths that can be requested between two configurations using the path planner.

Behavior-Based Languages

Behavior based programming languages provide an alternative to the procedural languages. Behavior based programming defines how the robot should react to some stimulus or event rather than following a procedural description. The idea behind behavior-based programming is to supply a set of behaviors that independently work to accomplish their goals, but together allow the robot to accomplish larger tasks. Behavior-based programming employs a hierarchical system of behaviors specifically written to perform an action based on a set of triggers (cruise, bumper escape, avoid, home, etc..). As the complexity of the overall system increases, new behaviors can be added without changing existing ones.

Functional Reactive Programming (FRP) is a good example of behavior based programming. FRP reacts to both analog and discrete signals. Yampa [30] and Frob [31] are example are two recent extensions of the FRP architecture. The advantage of FRP is that it is much more code efficient in comparison to procedural languages. In Yampa. for example, it is possible to write a wall-following algorithm with just 8 lines of code.

3.1.2 Graphical Programming Environments

Graphical programming environments provide an alternative to text-based programming environments. Even-tough graphical programming environments are closer to automatic programming systems, they are still regarded as manual programming. This is because, the user still needs to manually design the program flow and actions. Graphical systems utilize graphs, flow charts or diagrams to provide means for programming the robot. Small interdependent modules are connected to each other to create procedural flow or behaviors.

Lego's Mindstorm NXT [32] products provide a very successful flowchart based programming environment (Fig. 3.3). Since it is targeted at children, it

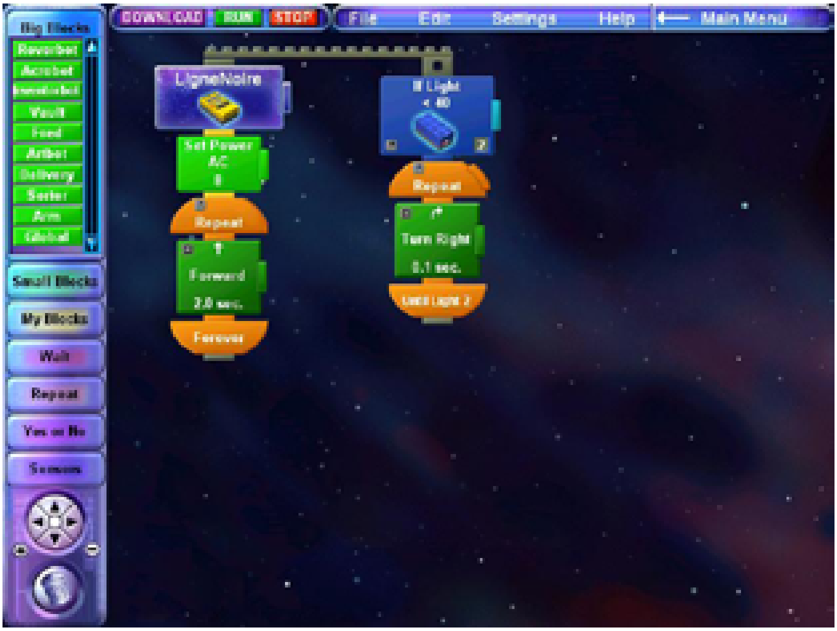

Figure 3.3: Lego Midstorm programming environment

is very simple in its design. In the programming environment iconic building blocks representing low-level functions are stacked together to produce a sequence of actions. It is also possible to create macros with in the programming environment. The generated sequence of commands can either be executed as the main process of the robot or mapped as a behavior when a certain sensor is triggered. A similar approach developed by Bischoff *et al.* [33] has been used to program industrial robots. In their system the user joins iconic low-level functions to reconfigure the robot to perform the required tasks. Usability tests show that both experts and novice users found the graphical system easier for handling robots.

3.2 Automatic Programming Systems

3.2.1 Learning Systems

In learning systems, the robot learns by inductive inference from user provided examples and self exploration [34]. First the robot watches and observes the user through a range of sensors and then tries to imitate the user. Billard and Schaal created a hierarchy of neural networks developed for learning the motion of human arm in 3D space [35]. Weng and Zhang proposed a robot that can learn simple tasks and chain them together to form larger and complex behaviors [36].

3.2.2 Programming by Demonstration

Programming by demonstration (PbD) is also a common way of tutoring robots for trajectory oriented tasks such as arc welding or gluing [1]. PbD started about 30 years ago with the development of industrial robots and has grown importantly in the last decade with the advances in computer science and sensor technology. Traditional PbD systems use a teach-pendant to jog the robot to the desired position. This position is recorded and a sequence of these positions is used to generate a robot program that will move the robot through a certain path. This method has been the industry standard for many years. In research this traditional ways of guiding/teleoperating the robot was progressively replaced by more user-friendly and intuitive interfaces [37], such as vision [38, 39, 40], data gloves [41], laser range finder [40] or kinesthetic teaching (ie. by physical guiding the robot's arm through motion)[42, 43, 44]. Kinesthetics provide a very rapid way of teaching new paths to robots, especially when used in assembly. Myers *et al.* used programming by demonstration to teach the robot subtasks which are then grouped into sequential tasks by the programmer [45].

Over the years, research and applications moved from simply copying or imitating the demonstrations to generalizing across a set of demonstrations. Münch et al. suggested the use of machine learning (ML) techniques to recognize elementary operators, thus defining a discrete set of basic motor skills. In their work, they also issued how to generalize a task, how to reproduce a skill in a completely novel situation, how to refine the reproduction attempt, and how to better define the role of the teacher during learning [46]. There are two different approaches for skill representation; A low-level representation of the skill can be seen as nonlinear mapping between sensory and motor inputs; trajectory encoding is an example of low-level skills. By contrast, high-level representation of a skill decomposes the skill into a sequence of elementary action and perception units also referred as symbolic encoding [37]. A significant portion of the work done in the PbD field uses symbolic representation of both the learning and the encoding of skills and tasks [46, 47].

3.2.3 Instructive Systems

In the case of instructive systems, series of instructions are given to the robot. This type of programming is best suited for executing series of tasks that the robot is already trained for. Using speech to instruct a robot provides a natural and intuitive way. Lauria *et al.* use a speech based natural language input to navigate a mobile robot to different locations via specified routes [48]. Brick and Scheutz provide an incremental framework where the robot can act upon sufficient information to distinguish the intended referent from perceivable alternatives, even when this information occurs before the end of the syntactic constituent [49]. Hand gestures are also used as input; Voyles and Khosia integrated a gesture based set of commands into a programming by demonstration framework [50]. Strobel *et al.* use static gestures to direct the attention of the robot to a specific part of the scene [6]. Combining the two modalities can even provide improved robustness [51]. McGuire *et al.* make use of speech and static gestures in order to draw the robot's attention to the object of interest and make it grasp this object [19].

Chapter 4

Results

4.1 Contributions

The contributions presented in this thesis can be divided into two parts:

4.1.1 General Multimodal Framework

It is believed that using multiple modalities makes a message less prone to being misunderstood, depending on the type of data; therefore multimodality can help to reduce ambiguities in human-computer interaction [9]. There have been numerous efforts to implement multimodal interfaces for computers and robots [11, 12, 13, 14]. Yet there is no general standard framework for developing multimodal interfaces. In order to design and implement such interfaces efficiently, we propose a general framework. The proposed framework in Paper C is designed to perform natural language understanding, modality fusion and semantic analysis through an incremental pipeline.

The framework also employs a new grammar definition language which is called COactive Language Definition (COLD).COLD is responsible for multimodal grammar definition and semantic analysis representation. COLD is used to (1) generate separate grammars for each modality, (2) define the fusion pattern, (3) define the semantic variables and calculations and (4) define dialog patterns and dialog turns. This means that COLD affects the whole process from the processing of inputs to modality fusion, semantic analysis and dialog management.

The framework is also an incremental system. The incremental nature of the system allows to start processing of input words or signals from other modalities as they are received before the sentence is complete. Parsing, modality fusion, semantic meaning generation and execution are all performed through this incremental pipeline. With an incremental system it is easy to build a response to user inputs before the sentence is finished. In the HCI/HRI domain, incremental processing helps to improve the response times of computer systems. Multimodal systems should be very responsive, because they would otherwise require the repetition of commands from the user, more ambiguity in the recognition process as well as user annoyance

4.1.2 Object-Based Programming Scheme

Programming industrial robots is not an easy task for a person who doesn't have previous experience. One of the reasons is that even though we occupy the same physical space as the robots, the intrinsic representation of the world is different for humans and robots. Humans represent the world around them by describing objects and spatial relations between these objects in a natural language. Robots, by contrast, work in abstract numeric coordinate systems which are not intuitive to us. Paper A, B and D aim to find a mapping between our object-based representation of the world and the robots' numeric representation.

The simulation environment presented in Paper B provides a procedural language API for ABB industrial robots, making it possible to program them through C# [52]. With an extended version of this API it is also possible to use Prolog [53], which is a general-purpose logic programming language, to program the robot. Having support for generic procedural and logic programming languages through simulation environment, makes it possible to automatically generate error free RAPID code for ABB robots [29]. Calls from C# are tested for reachability along a path. Collision free path planning from point A to point B can be performed. The motivation for using Prolog is twofold. Firstly, it is used to develop an initial prototype for natural language processing that is later on replaced by COLD framework. Secondly, using backtracking makes it possible to generate long list of RAPID commands from simple PROLOG commands. All these properties of the simulation environment makes it a useful and reliable research tool.

Paper B explores the use of natural language backed with spatial relations in an industrial environment. In the paper, we demonstrate a high-level language in order to command an industrial robot for simple pick-and-place applications.

The proposed high-level language allows the user to give high-level commands to the robot to manipulate the objects. The proposed language can also handle attributes of the objects in the environment, such as shapes, colors, and other features in a natural way. It is also equipped with functions for handling spatial information, enabling the user to be able to relate objects spatially to static objects. Paper B aims to command a robot to define which operations or tasks to be performed on a selection of objects.

The language described in Paper B works at a very high-level. The robot knows what to do and the user only instructs the robot to do the operation on a set of objects that pass a criterion. A possible example is “put all the blue objects into the palette”. The user has no freedom of choosing which blue objects go in to which locations in the palette. In order to address this problem, Paper D lowers the level of abstraction of commands, giving the user the ability to determine which object goes where. Paper D utilizes the general multimodal framework backed up with an augmented reality interface. With the camera mounted on the gripper of the robot, the user can see through the eyes of the robot and select objects and drop locations for these objects.Paper D addresses to describe how a task should be performed.

Combining the proposed general multimodal framework with the object-based programming scheme open new possibilities for creating interactive systems for robot programming. One of the these possibilities injunction with the goals of the thesis is to create a instructive programming environment that can easily be used by people who does not have expert knowledge in robot programming.

4.2 Overview of Papers

4.2.1 Paper A

Interacting with Industrial Robots Through a Multi-modal Language and Sensory Systems, Batu Akan, Baran Çürüklü, Lars Asplund, In proceedings of the 39th International Symposium on Robotics (ISR), Seoul, Korea, October, 2008

Summary: In this paper we propose a theoretical model for the framework, that we build the system on. The paper discusses the issues about difficulties in programing of industrial robots and why these robots are not used in small and medium sized enterprises (SMEs). As a solution we propose a high level language that acts a mediator between a user and a robot.

My contribution: I was the main author of this paper; however, all authors contributed to the idea and the writing process.

4.2.2 Paper B

Object Selection using a Spatial Language for Flexible Assembly, Batu Akan, Baran Çürüklü, Giacomo Spampinato, Lars Asplund, In Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'09), Mallorca, Spain, September, 2009.

Summary: In this paper we propose a limited natural language that utilizes spatial terms to hide the complexities of robots programming. We use Gaussian kernels to represent spatial regions such as "left" or "above". We also introduce our robot simulation environment where we check for reachability and collisions. The simulation environment also provides application programmers interfaces for procedural and behavioral programming in C# and Prolog languages.

My contribution: I was the main author of this paper contributing with the development of the system, together with the programming interfaces. The co-authors contributed with technical aspects, such as inverse kinematics for the robot and valuable feedback on the overall paper.

4.2.3 Paper C

A General Framework for Incremental Processing of Multimodal Inputs, Afshin Ameri E., Batu Akan, Baran Çürüklü, Lars Asplund, In Proceedings of the 13th International Conference on Multimodal Interaction (ICMI'11), Alicante, Spain, November, 2011.

Summary: In this paper we propose a framework for the rapid development of multimodal interfaces. The framework is designed to perform modality fusion and semantic analysis through an incremental pipeline. The incremental pipeline allows for semantic analysis and meaning generation as the inputs are being received. The framework also employs a new grammar definition language which is called COactive Language Definition (COLD). COLD is responsible for multimodal grammar definition and semantic analysis representation. This makes it easy for multimodal application developers to view and edit all the required resources for representation and analysis of multimodal inputs at one place and through one language.

My contribution: As the second author of this paper, I contributed with the idea and formalization of COLD,the partial development of the system, experiments and the editing of the paper

4.2.4 Paper D

Intuitive Industrial Robot Programming Through Incremental Multimodal Language and Augmented Reality, Batu Akan, Afshin Ameri E., Baran Çürüklü, Lars Asplund, In proceedings of the IEEE International Conference on Robotics and Automation (ICRA'11), Shanghai, China, May, 2011.

Summary: In this paper, we propose to use an incremental and multimodal natural language, which we developped in Paper C, in combination with our simulation environment and augmented reality. A view of the working environment is presented to the user through a unified system. The system overlays visuals through augmented reality to the user and also receives inputs and voice commands through our high level multimodal language. The proposed system architecture makes it possible to manipulate, pick or place the objects in the scene. Such a language shifts the focus of industrial robot programming from a coordinate-based programming paradigm to an object-based programming scheme.

My contribution: I am the main author of this paper and contributed with the development of the AR module, the experimental setup and the writing of the paper. The co-authors contributed with technical aspects and valuable feedback on the overall paper.

Chapter 5

Conclusions

In this thesis we propose general multimodal framework and a object-based programming scheme for interacting with industrial robots by using natural means of communication. The proposed system provides an alternative to well-established robot programming methods which require considerable larger amount of time, and perhaps more importantly, expert knowledge. A system which is easy to learn and use enables large numbers of users to benefit from that technology. Minimizing the need of expert knowledge is very beneficial for the industry, especially for small and medium-sized enterprises(SMEs). Providing such a solution to the SMEs might encourage them to invest in robot automation which in return will boost productivity. It is not only SMEs that can benefit from the easy programming of industrial robots; integrators, who develop robotic automation solutions for various companies can do so too. Developing and delivering solutions to their customers at the fraction of the time it took before would increase their competitiveness in the market.

5.1 Future Work

This thesis opens up possibilities to conduct further research in certain areas that have not thoroughly addressed.

The current work so far addresses the interaction between the robot and the user. In a way the robot understands the user's intentions, but it has no sense of the environment. Object recognition and localization are essential abilities for robots in general to work in unstructured environments. From the point of view of this thesis, object localization abilities will also enrich the interaction process between the robot and the user. In the future we plan to add a camera system capable of object recognition and object localization.

In our initial experiments, users have found the system easy to learn and to use; they also have reported that it was "fun" to use compared to the traditional programming method. However, more experiments, especially in real industrial workplaces and with user groups having different backgrounds are necessary to be carried out.

The current natural grammar is very limited in size. As the grammar grows upon requests from the industry, it is necessary to analyze the impact of growth upon the COLD framework. Another task regarding COLD framework would be to introduce a context analyzer to improve the robustness of the framework as the size of the grammar gets bigger.

Bibliography

[1] R D Schraft and C Meyer. The Need For an Intuitive Teaching Method For Small and Medium Enterprises. International Symposium on Robotics, May 2006.

[2] Michael a. Goodrich and Alan C. Schultz. Human-Robot Interaction: A Survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[3] Thomas B. Sheridan and William L. Verplank. Human and Computer Control of Undersea Teleoperators. Technical report, 1978.

[4] T Zhang, V Ampornaramveth, MA Bhuiyan, Y Shirai, and H Ueno. Face and gesture recognition using subspace method for human-robot interaction. *Advances in Multimedia*, pages 369–376, 2005.

[5] R.M. Voyles and P.K. Khosla. Gesture-based programming: a preliminary demonstration. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, (May):708–713, 1999.

[6] M Strobel, J Illmann, B Kluge, and F Marrone. Using spatial context knowledge in gesture recognition for commanding a domestic service robot. pages 468–473. Robot and Human Interactive Communication, 2002, 2002.

[7] Paul Schermerhorn, Matthias Scheutz, and Charles R. Crowell. Robot social presence and gender: do females view robots differently than males? In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 263–270, 2008.

[8] Timothy Brick and Matthias Scheutz. Incremental natural language processing for HRI. *Proceeding of the ACM/IEEE international conference on Human-robot interaction - HRI '07*, page 263, 2007.

[9] S. Oviatt. Ten myths of multimodal interaction. *Communications of the ACM*, 42(11):74–81, 1999.

[10] Richard A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *International Conference on Computer Graphics and Interactive Techniques*, volume 14, page 262, 1980.

[11] Kai-yuh Hsiao, Soroush Vosoughi, Stefanie Tellex, Rony Kubat, and Deb Roy. Object schemas for responsive robotic language use. *Proceedings of the 3rd international conference on Human robot interaction - HRI '08*, page 233, 2008.

[12] D. Mori, S. Matsubara, and Y. Inagaki. *Incremental parsing for interactive natural language interface*. IEEE, 2001.

[13] Michael Johnston and Srinivas Bangalore. Finite-state multimodal parsing and understanding. *Proceedings of the 18th conference on Computational linguistics -*, pages 369–375, 2000.

[14] David Schlangen and Gabriel Skantze. A general, abstract model of incremental dialogue processing. In *European Chapter Meeting of the ACL*, pages 710–718, 2009.

[15] Michael Johnston. Unification-based multimodal parsing. *Annual Meeting of the ACL*, page 624, 1998.

[16] C. Fuegen, Hartwig Holzapfel, and Alex Waibel. Tight coupling of speech recognition and dialog management-dialog-context dependent grammar weighting for speech recognition. In *Eighth International Conference on Spoken Language Processing*, pages 2–5. Citeseer, 2004.

[17] G.J.M. Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, and Nick Hawes. Incremental, multi-level processing for comprehending situated dialogue in human-robot interaction. In *Language and Robots: Proceedings from the Symposium (LangRo2007)*, pages 55–64, 2007.

[18] S. Iba, C.J.J. Paredis, and P.K. Khosla. Interactive multi-modal robot programming. *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, pages 161–168, 2004.

[19] P C McGuire, J Fritsch, J J Steil, F Roethling, G A Fink, S Wachsmuth, G Sagerer, and H Ritter. Multi-Modal Human-Machine Communication for Instructing Robot Grasping Tasks. pages 1082–1089. nternational Conference on Intelligent Robots and Systems (IROS), 2002.

[20] Osamu Sugiyama, Takayuki Kanda, Michita Imai, Hiroshi Ishiguro, Norihiro Hagita, and Yuichiro Anzai. Humanlike conversation with gestures and verbal cues based on a three-layer attention-drawing model. *Connection Science*, 18(4):379–402, December 2006.

[21] R R Murphy. Humanrobot interaction in rescue robotics. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, 34(2):138–153, 2004.

[22] Jason S McCarley and Christopher D Wickens. Human factors implications of uavs in the national airspace. *Human Factors*, (April), 2005.

[23] Jean Scholtz. Theory and evaluation of human robot interactions. *System Sciences, 2003. Proceedings of the 36th*, 2003.

[24] Z. Pronk and M. Schoonmade. *Mission preparation and training facility for the European Robotic Arm (ERA)*. NLR TP // Nationaal Lucht- en Ruimtevaartlaboratorium. Nationaal Lucht- en Ruimtevaartlaboratorium, 1999.

[25] Takayuki Kanda, Takayuki Hirano, Daniel Eaton, and Hiroshi Ishiguro. Interactive robots as social partners and peer tutors for children: A field trial. *Human-Computer Interaction*, 19(1):61–84, 2004.

[26] Robin R. Murphy. National science foundation summer field institute for rescue robots for research and response (r4). *AI Magazine*, pages 133–136, 2004.

[27] G. Trafton. Experimental desing in hri, 2007.

[28] Geoffrey Biggs and B. MacDonald. A survey of robot programming systems. In *Proceedings of the Australasian conference on robotics and automation*. Citeseer, 2003.

[29] ABB Flexible Automation, 72168, Vasteras, SWEDEN. *RAPID Reference Manual 4.0*.

[30] Paul Hudak, Antony Courtney, Henrik Nilsson, and John Peterson. Arrows, robots, and functional reactive programming. In *Advanced Functional Programming, 4th International School, volume 2638 of LNCS*, pages 159–187. Springer-Verlag, 2002.

[31] J Peterson, G D Hager, and P Hudak. A language for declarative robotic programming. *Proceedings 1999 IEEE International Conference on Robotics and Automation Cat No99CH36288C*, pages 1144–1151, 1999.

[32] Lego mindstorms. `http://mindstorms.lego.com/en-us/Default.aspx`. Accessed: 13/02/2012.

[33] Rainer Bischoff, Arif Kazi, Markus Seyfarth, Arifkazi Kuka-roboter De, and Industriegebiet Der B. The MORPHA Style Guide for Icon-Based Programming. *Robotics*, 2002.

[34] G Biggs and B MacDonald. A Survey of Robot Programming Systems. Australasian Conference on Robotics and Automation, December 2003.

[35] A Billard and S.Schaal. Robust learning of arm trajectories through human demonstration. pages 734–739. Intelligent Robots and Systems, 2001.

[36] Y Zhang and J Weng. Action chaining by a developmental robot with value system. pages 53–60. International Conference on Development and Learning, 2002.

[37] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. *Handbook of Robotics : Robot Programming by Demonstration*, chapter 59. Springer, January 2008.

[38] Y Kuniyoshi, M Inaba, and H Inoue. Teaching by showing: Generating robot programs by visual observation of human performance. *International Symposium on Industrial Robots*, 1989.

[39] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822, 1994.

[40] Sing Bing Kang and Katsushi Ikeuchi. A Robot System that Observes and Replicates Grasping Tasks. In *The Fifth International Conference on Computer Vision*, pages 1093–1099, June 1995.

[41] C P Tung and A C Kak. Automatic learning of assembly tasks using a DataGlove system. In *IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, Washington, DC, USA, 1995. IEEE Computer Society.

[42] S Calinon, F Guenter, and A Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems,Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, pages 286–298, 2007.

[43] M Ito, K Noda, Y.Hoshino, and J Tani. Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model. *Neural Networks*, 19(3):323–337, 2006.

[44] T Inamura, N Kojo, and M Inaba. Situation recognition and behavior induction based on geometric symbol representation of multimodal sensorimotor patterns. In *Intelligent Robots and Systems,*, pages 5147–5152, Beijing, China, October 2006.

[45] D R Myers, M J Pritchard, and M D J Brown. Automated programming of an industrial robot through teach-by showing. pages 4078–4083. International Conference on Robotics and Automation, 2001.

[46] J Kreuziger, M Kaiser, and R Dillmann. Robot programming by demonstration (rpd) - using machine learning and user interaction methods for the development of easy and comfortable robot programming systems. In *In Proceedings of the 24th International Symposium on Industrial Robots*, pages 685–693, 1994.

[47] Michael Pardowitz, Steffen Knoop, and Ruediger Dillmann. Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART B: CYBERNETICS*, 37(2):322–332, 2007.

[48] S Lauria, G Bugmann, T Kyriacou, and E Klein. Mobile robot programming using natural languages. *robotics and Autonomous Systems*, 38(3):171–181, March 2002.

[49] T Brick and M Scheutz. Incremental Natural Language Processing for HRI. HRI'07.

[50] R M Voyles and P K Khosia. Gesture-Based Programming: A Preliminary Approach.

[51] S Iba, C J J Paredis, and P Khosla. Interactive Multi-Modal Robot Programming.

[52] C# language specification. `http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf`. Accessed: 13/02/2012.

[53] Jan Wielemaker. An overview of the {SWI-Prolog} Programming Environment. In Fred Mesnard and Alexander Serebenik, editors, *Proceedings of the 13th International Workshop on Logic Programming Environments*, pages 1–16, Heverlee, Belgium, December 2003. Katholieke Universiteit Leuven.