

Rapid Prototyping for Interactive Robots

Christoph Bartneck, Jun Hu
Department of Industrial Design
Eindhoven University of Technology
Den Dolech 2, 5600 MB Eindhoven, The Netherlands
{c.bartneck, j.hu}@tue.nl

Abstract. A major problem for the development of interactive robots is the user requirement definition, especially the requirements of the appearance and the behavior of the robot. This paper proposes to tackle the problem using the well-known rapid prototyping method from the software engineering, with detailed prototyping techniques adjusted for the nature of the robots and the tactile human-robot interaction.

1 Introduction

An increasing number of robots are being developed to directly interact with humans. This can only be achieved by leaving the laboratories and introduce the robots to the real world of the users. This may be at home, work or any other location that users reside. Several interactive robots are already commercially available, such as Aibo [1], SDR [2] and the products of the iRobot company [3]. Their applications range from entertainment to helping the elderly [4], operation in hazardous environments [3] and interfaces agents for ambient intelligent homes [5].

A major problem for the development of such robots is the definition of user requirements. Since the user have usually no prior experience it is impossible to simply interview them on how they would like their robot to be. To overcome this problem we would like to propose a rapid robotic prototyping method that directly relates to the well-known rapid software prototyping method.

2 Problem Definition

Humans have generally very limited experience interacting with robots. Their experiences and expectations are usually based on movies and books and therefore cultural dependent. The great success of robotic show events, such as RoboFesta [6] and Robodex [7], show that Japanese have a vivid interest in robots and consider them as partners to humans. Their positive attitude may be based on years of Anime cartoons, starting in the fifties with “Astro Boy” [8] and later in “Ghost in the Shell” [9] in which robots save humanity from various threats. In comparison, the attitude of Europeans is less positive. The success of movies such as “2001 Space Odyssey” [10], “Terminator” [11] and “The Matrix” [12] shows a deep mistrust towards robots. The underlying fear is that robots might take over control and enslave humanity.

Against such a cultural load, the appearance of robots is of major importance. It determines the attitude and expectations towards it. If, for example, it has a very human form people are likely to start talking to it and would expect it to answer. If the robot cannot comply with these expectations, the user will have a disappointing experience [13].

The problem lies in the exact definition of how such a robot should look and how it should behave. It is not possible to draw these requirements directly from users by interviewing them, since they have no prior experience and are largely influenced by culture as mentioned above. These difficulties should not tempt developers to ignore these requirements. Too often complex and expensive robots are developed without these requirements in mind [14]. Once such a robot is finished and showed to users it often conflicts with the user's needs and expectations and therefore does not gain acceptance.

We would like to describe typical challenges in the development process of robots before we propose a method to tackle them.

2.1 *Design challenges*

2.1.1 **Shape of the robot**

To reduce development costs many parts of a robot are based on standard components. They are usually stacked on top of each other and once it is operational a shell is build around it to hide it from the user. The shape of the robot is only considered after the technology is built [14]. Humans are very sensitive to proportions of anthropomorphic forms and therefore these robots are often perceived as mutants due to their odd shapes. The evolution of Honda's Asimo [15] is a positive example of integrating technology into a natural shape.

2.1.2 **Purpose**

Building robots is a challenging and exciting activity and some engineers build robots only for the fun of it. However, a robot in itself is senseless without a purpose. A clear definition of its purpose is necessary to deduct requirements which in turn increase the chance of the robot to become a success. An unfortunate example of a wrong purpose is Kuma [16]. It was intended to reduce the loneliness of elderly people by accompanying them during watching Television. It is unclear how that would increase human contact and hence tackle the root of the problem. A robot that improves communication [4] would be more successful.

2.1.3 **Social role**

The robot will show intentional behavior and therefore humans will perceive the robot to have a character [17]. Together with its purpose the robot plays a social role, for example the one of a butler. Such a role entails certain expectations. For example, you would expect a butler to be able to serve you drinks *and* food. A robot that would only be able to do the one and not the other would lead to a disappointing experience.

2.1.4 **Environment**

To be able to define the purpose and role of the robot it is necessary to consider the environment of the robot. What are the characteristics of it in terms of architectural and social structures? Sony's Aibo, for example, is not able to overcome even a small step and therefore

its action range is much more limited than an ordinary dog. This results into some frustration of its owners [18].

2.2 *Technical challenges*

Building real robots is hard. Robots are complex systems which rely on software, electronics and mechanical systems all working together. It requires the specialists to have the knowledge and skills that cover all these disciplines to bring a robot alive. Building robots such as the NASA's Mars Pathfinder [19] and the Honda's Asimo [15] is a mission impossible for an individual. A team of top scientists and engineers from different backgrounds need to work together closely in order to build such robots. These robots are therefore very expensive. Small robots such as Sony's Aibo are not cheap either. Although it is designed for home entertainment and many families can afford an Aibo, Sony must have paid a fortune for the designers, researchers and engineers in order to bring such a not-so-expensive product into the market. Most robots to date have been more like kitchen appliances.

2.3 *Empirical evaluation*

Another challenge in the development of interactive robots is the definition of measurable benefits. These benefits are influenced by the interaction with humans and hence difficult to measure. A wide range of methods and measurement tools are available from the human computer interaction research area [20] to help with the evaluation. Here we would only like to describe some of the common challenges that developers experience in the evaluation process.

The first is of course not to do any evaluation. Simply stating that a certain robot is fun to interact with is nothing more than propaganda. A first difficulty in the evaluation process is to clearly define what the actual benefit of the robot should be, how it can be measured and who the target user group is. Especially for vague concepts, such as "fun", it is difficult to find validated measurement tools. Furthermore, too few participants that are possibly even colleagues of the developers also often compromise the tests. A small and technical oriented group of participants does not allow a generalization across the target group. The participants need to come from the group of intended users. A problem with these users is the novelty effect [17, 13]. Interacting with a robot is exciting for users that have never done it before and hence their evaluation tends to be too positive.

3 **Solution**

The similar problems and difficulties described above exist also in software engineering. The similarity of the problems and difficulties lie in the software engineering suggests that the principles of rapid prototyping may also applicable in the development of interactive robots.

3.1 *Rapid prototyping in software engineering*

The similar problems and difficulties described above exist also in software engineering. The similarity of the problems and difficulties in software engineering suggests that the principles of rapid prototyping may also be applicable in the development of interactive robots.

3.2 Rapid prototyping in software engineering

Requirements definition is crucial for successful software development. Obtaining user requirements solely by interviewing the users is difficult and in many cases unreliable. Some users have expectations for computers that is either so high that they lead to requirements are more stringent than what is really needed, or so low that they hide the requirements from the developers. With limited prior experience of existing software solutions, the users can not specify the requirements until they experience some of the solutions. Lacking of the user's domain knowledge, the software developers also often find it hard to explain to the users what is feasible until they manage to visualize their ideas.

To tackle this problem, the rapid prototyping model is introduced against the framework of the conventional water-fall life cycle models [21, 22, 23]. The rapid prototyping model strives for demonstrating functionality early in software development process, in order to draw requirements and specifications. The prototype provides a vehicle for the developers to better understand the environment and the requirements problem being addressed. By demonstrating what is functionally feasible and where the technical weak spots still exists, the prototype stretches their imagination, leading to more creative and realistic inputs, and a more forward-looking system.

Before elaborating the details of rapid robotic prototyping, we first have a look at what makes it different from software prototyping and why it is worth a separate discussion.

3.3 Difference between robotic prototyping and software prototyping

The first difference is that the target system of robotic prototyping is a robot, which has its physical existence in the 3D world, while a software system is just an artifact that exists digitally in a virtual space. One of the most important goal of rapid robotic prototyping is to investigate the user requirements of the physical appearance and behavior, hence implementing a robotic prototype is not just programming to give it intelligence, but more importantly, to build its physical embodiments. Software prototyping often uses existing software packages, modules and components to accelerate the process, while robotic prototyping often needs electronic and mechanical building blocks.

The physical embodiments of the interactive robots encourage naturally the tactile human-robot interaction. The user and the robot exchange the tactile information, ranging from force, texture, gestures to surface temperature. The tactile interaction is seldom necessary in most of the software systems, the interfaces of which are often confined in a 2D screen. The 2D interfaces of such could be easily prototyped using low-fidelity techniques such as paper mock-ups, computer graphics and animations, whereas these techniques are not suitable for prototyping the tactile interaction that is essential to most of the interactive robotic systems.

Another differences lies in the intermediate prototypes. The prototypes built during the software prototyping can possibly be evolving or growing into a full functioning system. This is so called evolutionary prototyping. During the process of evolution, a software design emerges from the prototypes. Rapid robotic prototyping proposed in this paper is only for quickly eliciting the user requirements. To make the process faster, the efficiency, reliability, intelligence and the building material of the robot is less of importance. Hence the prototypes produced in robotic prototyping are not intended to be ready for industrial reproduction.

3.4 Rapid robotic prototyping techniques

Keeping these differences in mind, we are now ready to review the often used prototyping techniques and propose the corresponding methods in robotic prototyping, following two dimensions of robotic prototyping: Horizontal prototyping realizes the appearance but eliminates depth of the behavior implementation, and vertical prototyping gives full implementation of certain selected behaviors. If the focus is on the appearance or the interface part of the robot, horizontal prototyping is needed and it results in a surface layer that includes the entire user interface to a full-featured robot but with no underlying functionality; if prototyping is to explore the details of certain features of the robot, vertical prototyping is necessary in order to be tested in depth under realistic circumstances with real user tasks.

3.4.1 Scenarios

Without any horizontal and vertical implementation, scenarios are the minimalist, and possibly the easiest and cheapest prototype in which only a single interaction session is described, encapsulating a story of a user interacting with robotic facilities to achieve a specific outcome under certain circumstances over a time interval. As in software prototyping, scenarios can be used during the early requirement analysis to inspire the user's imagination and feedback without the expense of constructing a running prototype. The form of the prototype can be a written narrative, or detailed with pictures, or even more detailed with video.

3.4.2 Paper mock-ups

In software prototyping, paper mock-ups are usually based on the drawings or printouts of the 2D interface objects such as menus, buttons, dialog boxes and their layout. They are turned into functioning prototypes by having a human "play computer" and present the change of interface whenever the user indicates an action. The system is horizontally mocked up with low fidelity technique, vertically faked with human intelligence.

Although they are also very useful in robotic prototyping to make the scenarios interactive, paper mock-ups in robotic prototyping are of even lower fidelity. In software prototyping, 2D interfaces are mocked up with the 2D objects on paper. To reach the same fidelity level in robotic prototyping, the 3D robot should be mock-up with a box of sculptures or 3D "print-outs" instead of a pile of drawings, which is time-consuming and expensive, though technically possible. Prototyping the 3D robotic appearance and behavior on 2D paper is just like prototyping the 2D software interface on a 1D line. Too much fidelity is lost. This argument has led us to mock-up the robots using robotic kits.

3.4.3 Mechanical Mock-ups

To keep the horizontal fidelity to a certain level, it is necessary to mock up the physical 3D appearance and mechanical structure of the robot. Robotic kits such as Evolution Robotics [24] and LEGO Mindstorms [25] are good tools to build such mock-ups. These kits come with not only common robotics hardware such as touch sensors, rotation sensors, temperature sensors, step motors and video cameras, but also mechanical parts such as beams, connectors and wheels, and even ready made robot body pieces and joints. One can assemble a robot

easily and quickly according to the needs, and can expect less workload of mechanical and electronic design.

To make a mock-up, only mechanical parts are needed to build up the skeleton. With some simple clothing, the robot appearance can be built with a higher fidelity. The behavior of the robot can be faked up by a human manipulating the prototype like a puppet show, according to the designed interactive scenario.

3.4.4 Wizard of OZ

The person who “plays computer” using paper mock-up or who “operates the puppet” using the mechanical mock-up can be a disturbing factor since the user may feel interacting with the person, not the robot. One way to overcome this is using the Wizard of OZ technique. Instead of operating the robot directly, the person plays as a “wizard” behind the scene and controls the robot remotely with a remote control, or from a connected computer. If controlled with a remote control, the “wizard” takes the role of the sensors by watching the user in a distance, and the role of the robot’s brain by make decisions for the robot. If controlled from a connected computer, the “wizard” only acts as the brain.

The prototype needed is more complicated than the mechanical mock-up. It has to be equipped with electronics such as power supply, sensors, actuators, a processor to control sensors and actuators, and a connection to a remote computer.

To keep it easy and simple, we have been using these robotic components from the Lego Mindstorms in our projects. These components are shaped nicely to fit with other mechanical parts, so that we can upgrade the mechanical mock-ups to “Wizard of OZ” prototypes easily without building from the ground.

3.4.5 Prototypes with high fidelity of intelligence

Using a human to take the role of the robot’s brain in above techniques pushes the vertical prototyping to an extreme – the robot tends to be too smart than it should be. In many cases we might want to prototype the intelligent behavior with a higher fidelity, for example, to investigate how the appearance and the behavior match each other, to discover where the technical bottlenecks are, and to observe how the robot interacts with its physical environment. In short, we might need to program the robot to enable its machine intelligence.

Many robots use a generic computer as its central processing unit. When it is too big to fit into the robots body, the computer is often “attached” to the robot via a wired or wireless connection. The advantage is that the programming environment is not limited by a specially designed robotic platform. The developer can choose whatever is convenient. The disadvantage is that the connection between the body and brain might become a bottleneck. The mobility of the robot is limited by the distance of the connection, and the performance is limited by the quality of the connection.

It would be better that the robot has its embodied processing unit that comes with an open and easy programming environment. This brings the Lego Mindstorms on the table again. From the set, the RCX is programmable, microcontroller-based brick that can simultaneously operate three motors, three sensors, and an infrared serial communications interface. The Lego enthusiasts have developed various kinds of firmware for it, which enables programming in Forth, C, and Java, turning the brick into an excellent platform for robotic prototyping [26].

Once the platform is selected, a good strategy of modeling and programming the robot helps to speed up the prototyping process. Considering the memory and processing power limitations of the RCX, we decided to use the behavior-based AI model and developed a design pattern for the robots in our projects. The behavior-based approach does not necessarily seek to produce cognition or a human-like thinking process. Instead of designing robots that could *think* intelligently, this approach aims at the robots that could *act* intelligently, with successful completion of a task as the goal. The similarity of the low level behavior of these robots leads us to develop an object-oriented design pattern that can be applied and reused.

4 Case studies

In this section we would like to shortly introduce some case studies in which the rapid robot prototyping method has been successfully used to gain insight into user requirements. Extended information for each of them is available at the given references.

4.1 *eMuu*

eMuu (Fig. 1) is intended to be an interface between an ambient intelligent home and its inhabitants [27]. To gain acceptance in the homes of users the robot needs to be more than operational. The interaction needs to be enjoyable. The embodiment of the robot and its emotional expressiveness are key factors influencing the enjoyability of the interaction. Two embodiments (screen character and robotic character) were developed using the *Muu* robot [28] as the base for the implementation. The sophisticated technology of *Muu* was replaced with Lego Mindstorms equipment. In addition, an eyebrow and lip was added to the body to enable the robot to express emotions. The evaluation of the robot showed that the embodiment did not influence the enjoyability of the interaction, but the ability of a robot to express emotions had a positive influence.



Figure 1: *eMuu*

4.2 *Tony*

Tony is designed for an interactive television show as a companion robot for the audience in the NexTV project [29]. The project is to investigate how the new interactive technologies such as MPEG-4, SMIL and MHP-DVB can influence the traditional television broadcasting. One of the important issues was that interactive media require new interface devices instead of computer keyboards or television remote controls. Following a user-centered approach, we showed an interactive movie to our target user group, children aged from 8 to 12, to investigate what and in which way they would like to be able to influence in the movie. One of the suggestions was to use a robot as a control device and they even made drawings to show what the robot should do and look like (Fig. 2a) [30]. Using Lego Mindstorms, the first version of *Tony* (Fig. 2b) was quickly built and showed to the users. The results from the user evaluation changed the role of *Tony* from a simple control device to a companion robot (Fig. 2c, 2d). *Tony* watches the show together with the user, performing certain behavior according to the requests from the show and influencing the show back according the instructions from the user [31, 32].

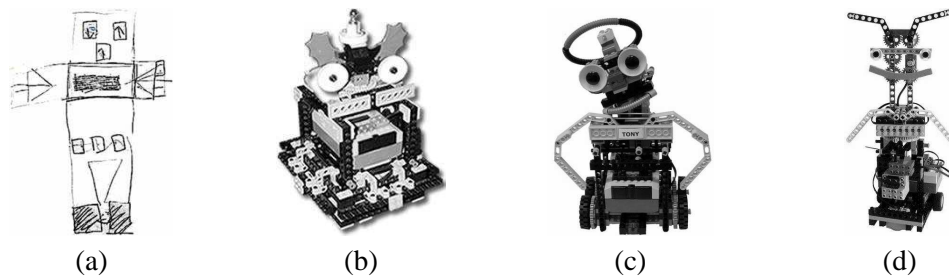


Figure 2: Tony

4.3 *LegoMarine*

LegoMarine (Fig. 3) is developed for an interactive 3D movie “DeepSea” in the ICE-CREAM project [33]. The 3D virtual world in the movie is extended and connected to the user’s environment by distributing sound and lighting effects, using multiple displays and robotic interfaces. The purpose of this movie is to investigate whether physical immersion and interaction will enhance the user’s experience. LegoMarine is used as the physical counterpart of a submarine in the virtual underwater. The user can direct the virtual submarine to navigate in the 3D space by tilting LegoMarine, or speed the submarine up by squeezing it. When the submarine hits something, LegoMarine also vibrates to give tactile feedback. The other behaviors of both are also synchronized, such as the speed of propeller and the intensity of the lights. At the beginning of the project, engineers in the project tried to build the robotic submarine by putting sophisticated electronics into a shell striped from a toy submarine, but later found out it is impossible to frequently change the shape and functions according to the user’s feedback. So the LegoMarine was born.

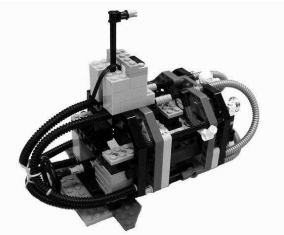


Figure 3: LegoMarine

4.4 *Mr. Point, Mr. Ghost and Flow Breaker*

These robots are developed for the researches on the gaming experiences. Mr. Point (Fig. 4a) and Mr. Ghost (Fig. 4b) were created for the well-known Pac Man game, as support robots respectively for the player and the ghosts in the game [34]. From this study the researchers learned that it is difficult to attract the attention of the users to the perception space formed by the physical agents. This observation led the researcher to explore physical and on-screen strategies to break game flow in an effective and user acceptable way. The prototype of Mr. Point was reused again, but renamed to Flow Breaker (Fig. 4c), with some modifications to fit onto the physical Pac Man maze [35]. These studies were conducted by the researchers who are also specialists in electronics and software. To build prototypes quickly, it is not necessary for them to use robotic kits to speed up the process. Still, these robots reflect the many principles of fast robotic prototyping: reusing existing components, building simple prototypes quickly and evaluating with real users.

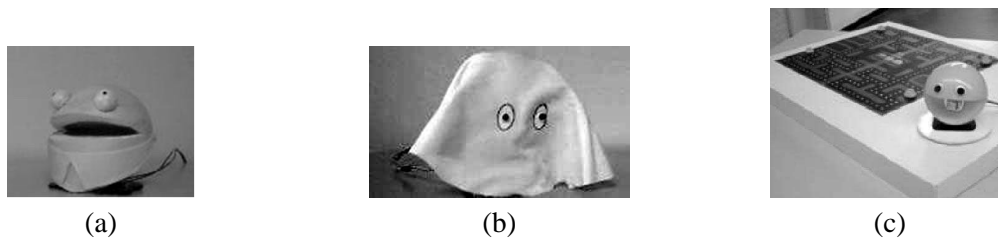


Figure 4: Game Support Robots

5 Conclusion

Rapid prototyping is a powerful method for defining the user requirements for interactive robots, as in software engineering. Many prototyping techniques from software engineering are still valid, but need to be adjusted for the nature of robots and the tactile interaction. We encourage non specialists to use robotic kits such as Lego Mindstorms to build robotic prototypes, which in our experience can simplify and accelerate the prototyping process.

In our projects, we also noticed that the Lego Mindstorms could not satisfy all our needs. Limited memory and speed of the processor, limited number of connected sensors and actuators, and poor infrared connections need a lot of improvements. It opens an opportunity for the industry to develop better robotic kits for prototyping and building interactive robots.

References

- [1] Sony, "Aibo," 1999, available from: <http://www.aibo.com>.
- [2] —, "Sony develops small biped entertainment robot," 2002, available from: <http://news.sel.sony.com/pressrelease/2359>.
- [3] iRobot, "iRobot," 2003, available from: <http://www.irobot.com>.
- [4] F. Gemperle, C. DiSalvo, J. Forlizzi, and W. Yonkers, "The hug: A new form for communication," in *Designing the User Experience (DUX2003)*. New York: ACM Press, 2003.
- [5] E. Aarts, R. Harwig, and M. Schuurmans, "Ambient intelligence," in *The invisible future: the seamless integration of technology into everyday life*, P. J. Denning, Ed. McGraw-Hill, Inc., 2001, pp. 235–250.
- [6] RoboFesta, "Robofesta (the international robot games festival)," 2002, available from: <http://www.robofesta.net/>.
- [7] Robodex, "Robodex: Robot dream exposition," 2002, available from: <http://www.robodex.org/>.
- [8] O. Tezuka, "Astro boy: The birth of astro boy," 2002, (VHS) by Ishiguro, N. and Tezuka, O. (Director), Yamamoto, S. (Producer): Manga Video.
- [9] K. Ito, "Ghost in the shell," 1992, (DVD) by Oshii, M. (Director), Ishikawa, M. (Producer): Palm Pictures/Manga Video.
- [10] S. Kubrick, "2001: A space odyssey," 1968, (DVD) by Kubrick, S. (Director), Kubrick, S. (Producer): Warner Home Video.
- [11] J. Cameron, "The terminator," 1984, (DVD) by Cameron, James (Director), Daly, J. (Producer): MGM.
- [12] A. Wachowski and L. Wachowski, "The matrix," 1999, (DVD) by Wachowski, Andy and Wachowski, Larry (Director), Berman, B. (Producer): Warner Home Video.
- [13] E. Diederiks, "Buddies in a box - animated characters in consumer electronics," in *Intelligent User Interfaces*, W. L. Johnson, E. Andre, and J. Domingue, Eds. Miami: ACM Press, 2003, pp. 34–38.

- [14] A. Bondarev, "Design of an emotion management system for a home robot," Post-master thesis, Eindhoven University of Technology, 2002.
- [15] Honda, "Asimo," 2002, available from: <http://www.honda.co.jp/ASIMO/>.
- [16] K. Hafner, "As robot pets and dolls multiply," 1999, available from: <http://www.telug.quebec.ca/psy2005/interac/robotsalive.htm>.
- [17] C. Bartneck, "emuu - an embodied emotional character for the ambient intelligent home," PhD thesis, Eindhoven University of Technology, 2002.
- [18] P. Menzel and F. D'Aluisio, *RoboSapiens*. Cambridge: MIT Press, 2000.
- [19] Nasa, "The mars exploration program," 2003, available from: <http://mars.jpl.nasa.gov/>.
- [20] J. Nielsen, *Usability Engineering*. San Francisco, California: Morgan Kaufmann Publishers, Inc., 1993.
- [21] A. M. Davis, "Rapid prototyping using executable requirements specifications," *ACM SIGSOFT Software Engineering Notes*, vol. 7, no. 5, pp. 39-42, 1982.
- [22] J. A. Clapp, "Rapid prototyping for risk management," in *the 11th International Computer Software and Applications Conference*. Tokyo, Japan: IEEE Computer Society Press, 1987, pp. 17-22.
- [23] Luqi and V. Berzins, "Rapidly prototyping real-time systems," *IEEE Software*, vol. 2, no. 5, pp. 25-36, 1988.
- [24] Idealab, "Robotics, robots, robot kits, oem solutions: Evolution robotics," 2003, available from: <http://www.evolution.com>.
- [25] LEGO, "Lego.com mindstorms home," 2003, available from: <http://mindstorms.lego.com/eng>.
- [26] B. Bagnall, *Core LEGO MINDSTORMS Programming: Unleash the Power of the Java Platform*. Prentice Hall PTR, 2002, no. 10/2/2002.
- [27] C. Bartneck, "Negotiating with an embodied emotional character," in *Design for Pleasurable Products*, J. Forzlizzi, B. Hamington, and P. W. Jordan, Eds. Pittsburgh: ACM Press, 2003, pp. 55-60.
- [28] M. Okada, "Muu: Artificial creatures as an embodied interface," in *ACM Siggraph 2001*, New Orleans, 2001, p. 91.
- [29] NexTV, "The nextv project homepage," 2001, available from: <http://www.extra.research.philips.com/euprojects/nextv>.
- [30] M. Bukowska, "Winky dink half a century later," Post-master thesis, Eindhoven University of Technology, 2001.
- [31] J. Hu, "Distributed interfaces for a time-based media application," Post-master thesis, Eindhoven University of Technology, 2001.
- [32] J. Hu and L. Feijs, "An adaptive architecture for presenting interactive media onto distributed interfaces," in *The 21st IASTED International Conference on Applied Informatics (AI 2003)*. Innsbruck, Austria: ACTA Press, 2003, pp. 899-904.
- [33] ICE-CREAM, "The ice-cream project homepage," 2003, available from: <http://www.extra.research.philips.com/euprojects/icecream/>.
- [34] M. de Graff and L. Feijs, "Support robots for playing games: the role of the player-actor relationships," in *People and Computers XVI - Memorable Yet Invisible*, X. Faulkner, J. Finlay, and F. Detienne, Eds. Springer-Verlag, 2002, pp. 403-417.
- [35] B. Eggen, L. Feijs, and P. Peters, "Linking physical and virtual interaction spaces," in *the International Conference on Entertainment Computing (ICEC)*, Pittsburgh, Pennsylvania, 2003.