Lino, the user-interface robot

B.J.A. Kröse[1], J.M. Porta[1], A.J.N. van Breemen[2], K. Crucq[2], M. Nuttin[3], and E. Demeester[3]

[1] University of Amsterdam,
Kruislaan 403, 1098SJ, Amsterdam, The Netherlands
{krose,porta}@science.uva.nl
[2] Philips Research,
Prof. Holstlaan 4, 5656AA, Eindhoven, The Netherlands
{albert.van.breemen,ko.crucq}@philips.com
[3] Katholieke Universiteit Leuven
Celestijnenlaan 300B, B-3001, Leuven (Heverlee), Belgium
{marnix.nuttin,eric.demeester}@mech.kuleuven.ac.be

Abstract. This paper reports on the development of a domestic user-interface robot that is able to have natural human interaction by speech and emotional feedback. Natural interaction with the user is achieved by means of a mechanical head able to express emotions. Additionally, our robot is aware not only of its position in the environment but also of the position and intentions of the users. The localization of the robot is achieved with an appearance-based localization method. To get information about the users, the robot is designed to operate in (and to cooperate with) an intelligent ambient that takes care of monitoring the users. The result is a service robot that departs from existing developments in the fields of interface and service robots that are mainly reactive and, thus, limited in functionality.

1 Introduction

In the last years an increasing effort is spent in research on service and entertainment robots which operate in natural environments and interact with humans. The Sony AIBO is an example of a robot which is meant to play with children: it has a perceptual system (vision, auditory, tactile), plays soccer, and can learn its own behavior [1]. NEC has developed "Papero", a *personal* robot which also is able to entertain the user but has more functionality: it serves as an interfacing with web-services and electronic equipment [18]. Even more functionality is present in various other service robots, such as robot-waiters [11], museum or exhibition robots [21, 3] or care-for-elderly robots [10], all examples of autonomous intelligent systems, operating in a real world.

Parallel to these robotic developments, a new paradigm in information technology is emerging, in which people are served by a digital environment that is aware of their presence and context, and is responsive to their needs, habits, gestures and emotions: ambient intelligence. Apart from the many challenges in networking technologies, perception and intelligence, there is an enormous challenge in the field of user interaction: is the user is going to talk to his or her toaster or coffee machine...? We think not.

As a part of the European project "Ambience" [13] we developed a domestic robot (see Figure 1). The robot must be some personification of the intelligent environment, and it must be able to show intelligent behavior, context awareness and natural interaction. The robot exploits the intelligent environment to get information about the user intentions, preferences, etc. In the other way around, the human user must be able to have a natural interaction with the digital world by means of the robot.

Fig. 1. The robot Lino.

Very important for the natural interaction is a nice look of the robot, and the possibility to express some emotional state. Many other robots use a (touch)screen interface, sometimes with an animated face [11, 7]. We decided to use a 'real' face, consisting of dynamic mouth, eyes and eyebrows since this makes the interaction more attractive and also more natural.

The development of software modules for different tasks is carried out by multiple developers. Therefore, we have implemented a dedicated software tool to support the developers. Using this tool, different software modules of the robot application, running on different operating systems and computers, can be connected/disconnected interactively at runtime by means of a graphical user interface. With this approach, integrating the different software components is a matter of "configuration" rather than programming.

The objective of this paper is to introduce the different modules developed for our robot, the software tools used to integrate them, and the preliminary results we have obtained so far.

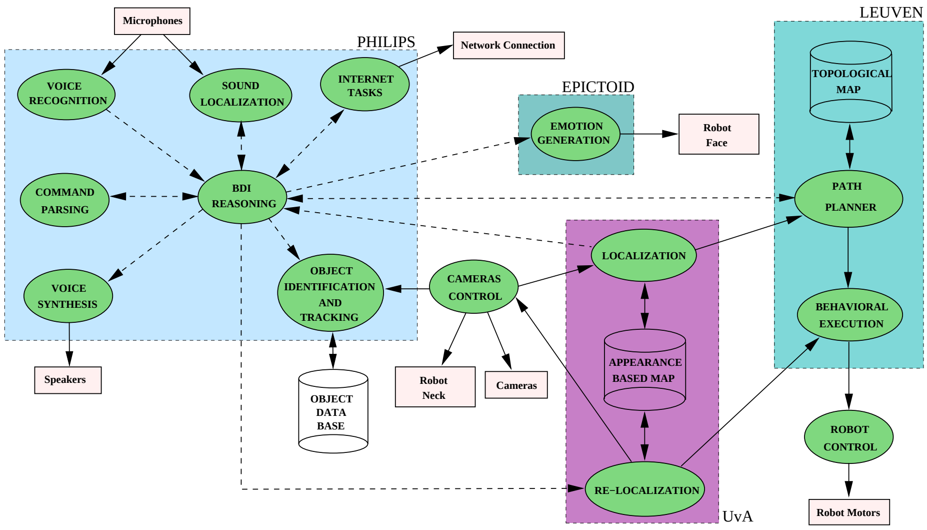

Fig. 2. Set-up of the software modules in Lino

2 Software Framework

The architecture is depicted in Figure 2. An efficient implementation and integration of all the different functional software components requires a dedicated software framework. We have developed a module-based software framework, called the *Dynamic Module Library* that services this purpose. The basic software construct is that of a *module* that has input and output ports, which can be connected to each other to exchange data. The framework meets the following requirements:

Runtime flexibility The possibility to change algorithms of modules at run-time, to extend the robot application with new modules at run-time, and to probe ingoing and outgoing data of modules at runtime.

Runtime robustness Stopping (or crashing) one or more modules of a running robot application should not result into an overall stopping (or crashing) of the robot application.

Runtime configurability The possibility to define at runtime the configuration of modules (that is, the connections between the modules) that make up the robot application.

Distribution The possibility to distribute the robot application over several host computers in order to enlarge the computational resources.

Portability Support for the most popular programming languages (C, C++, Java) and operating systems (Windows, Linux).

Modules are implemented separately from each other and are executed as individual processes in some operating system (both MS Windows and Linux are currently being supported). By using a registry, modules can discover and lookup each other at run-time. Once a module has found an other module, it can make connections to the ports of that other module. It is also possible to externally connect ports of modules. By means of the graphical user interface we can start/stop modules, connect/disconnect ports as well as probing ports. This way, a robot application consisting of modules can be configured at runtime which greatly enhances the interactivity of the development of the robot application.

In order to synchronize the tasks of all the modules, each module implements a particular task model. The model consists of five states in which a module can occur, namely `IDLE`, `RUNNING`, `PAUSED`, `SUCCEEDED` and `FAILED`. Each module can read as well as control the state of other modules. For instance, a Reasoning module can send an `execute` message to a Re-localization module to start this one up, and a `pause` message to a pathplanner module to pause its task.

3 User Awareness Module

We have implemented and tested one of the tasks of the robot called "Turn-to-speaker" behavior. This behavior basically determines the direction of a speaker in a 3D space and turns the head and body toward the speaker. The 3D speaker location estimation is determined by means of three mutually perpendicular microphone pairs. These microphones are mounted inside the head and are separated by a distance of 25 cm. Each microphone pair uses a stereo USB audio digitizer for the signal acquisition. We analyze the recorded signals to determine the difference in the time of flight of speech that arrives. The basic problem in this measurement is to get rid of the numerous acoustic reflections in the recorded speech signals. With an adaptive Filtered-Sum Beamformer and an optimization algorithm [4, 5] it is possible to determine the contribution of these reflections and to largely compensate for them in the recorded signals.

The location of the speaker is indicated in the local robot coordinate system by two angles, φ (horizontal plane), θ (tilt). The angle φ is used to turn the robot platform and θ is used to turn the head up to the speaker with the loudest voice. There is a problem when there are many speakers at the same level. In this case the system generates inconsistent values and we pause the turning of the robot. The system does not respond to random acoustic noise. It detects human voices by looking for harmonics (the pitch) in the recorded signal. With this technique we also want to explore “Follow-me” behavior: e.g. advancing small distances in the direction of someone speaking at regular intervals.

4 Emotion Engine

To generate the appropriate facial expressions and body language we have developed an emotion engine. This emotion engine autonomously reasons on the emotional state of the robot and is based on the psychological model of Ortony, Clore and Collins (OCC-model) [17]. The model has been applied successfully in other studies [9, 15]. The OCC-model reasons about the synthesis of emotions based on appraisal of consequences of events, of actions (self or of others) and of aspects of objects. The appraisal is evaluated by comparing the events (occurred or occurring) with the goals, the actions with the standards set and the appealingness of the objects with the attitudes set. Also the history is taken into account. Via a decision tree in which distinction is made between (1) consequences for self or others, (2) action of self or of others, (3) positive or negative aspects, (4) prospect relevant or irrelevant, (5) present or future, (6) desirable or undesirable a total of 22 possible emotional states result. The intensity of the emotion is determined by modelling functions for the desirability, the likelihood of occurrence, the appealingness, the praiseworthiness. Once the emotional state is determined the mapping to facial expressions (22) has to be achieved. This mapping has been carefully researched and devised by Epictoid.

5 Speech

We have implemented an interactive command and control dialogue system based on the SAPI of Microsoft. The speech recognition engine (SRE) is from Lernout & Hauspie (L&H) and the text to speech engine (TTS) is from AT&T. A simple dialogue management system has been devised which basically functions by transitions to different states, e.g. ”sleeping” → ”listing” → ”TV-control” → ”DVD-control” → ”idle”. After start-up the system is still in the ”sleeping” state. Issuing the command ” Lino wake up” will bring the system in the ”listening” state. In this state the system is ready to all kind of services, e.g. ”switch TV on”, ”set channel to ned1” or ”tell me what object do you see”. With the command ”go sleep” the system can be brought back into the sleeping mode. The system automatically switches to the ”idle” state after 20 sec.

Furthermore, the speech signal is also used to identify the current speaker. The algorithm we use is based on a Gaussian Mixture Model (GMM) [16]

The viseme output of the speech synthesizer is used to control and synchronize the lip movement during speech output. This lip synchronization contributes a lot to a more lively appearance.

6 Vision

Object detection, tracking and recognition is a very important capability for a robot. For the implementation of this module we have used the Inca-plus camera [12]. This camera is a stand-alone system and is normally used for machine vision. The complete image processing is locally done in the camera by means of two powerful processors; i.e. the Xetal for the pre-processing and the Trimedia for the actual image processing. In this way we have achieved a throughput rate of 10 Hz and higher. Only the CMOS-sensor chip has been mounted in the forehead of the robot. Currently, the robot is able to detect and recognize simple objects by means of the color and the overall shape (aspect ratio). Once detected, the eyes track the moving object which gives visual feedback that is appreciated very much by user. By means of speech output Lino can report what object it sees. Currently, we are implementing face detection and recognition.

7 Localization and Navigation

In order to navigate to a desired location, the robot must be able to *localize* itself, it has to *plan* a path and it has to *avoid obstacles* while following the path. For estimating its position, the robot has to compare its sensoric measurements with an internal representation. Of course it can also measure its wheel revolutions (odometry), but this is subjected to large errors in the long term. Consequently, additional sources of information must be used to determine the robot's position. For the Lino robot we use a vision system for this.

Our localization method is an 'appearance-based' method, which departs from a training set of images taken at known positions. The image set is transformed to a set of linear feature vectors and stored on the robot. For a robust localization we use a Markov procedure, where the 'belief' in the location of the robot is updated using new observations. Since the robot can turn its head, we actively acquire the best observations. A description of the probabilistic model, the Monte Carlo implementation and the active vision strategy is given in [14, 23, 19]. An additional advantage of using a stereo vision system is that it can provide depth maps that are less sensitive to change in illumination than usual intensity images. Recently we combined both modalities [20].

The objective of the navigation module is to drive the robot to the desired positions avoiding collision with obstacles. To accomplish this objective, a hybrid architecture is used in the navigation module. There are two internal modules: the planner and the behavioral execution module.

The planner generates subgoals from the robot's current location to the global goal position using a map. In previous work, a computationally efficient planner was developed based on the Wave Front Distance Field (WFDF) algorithm; see [22] for more details. This planner finds the shortest path to the goal and calculates subgoals on this path. The straight path between two consecutive subgoals is obstacle free. In a final step, subgoals that are close to each other are merged. This avoids that subgoals are too close together, which is not desirable

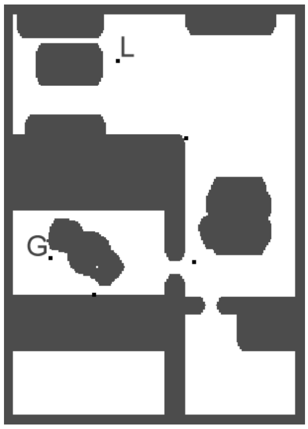

Fig. 3. The planner uses a map to generate subgoals (the small dots in the figure) from the current location L to the goal position G.

in behavioral execution. The efficiency of the algorithm allows to re-plan approximately four times every second, coping with the robot getting off course in front of obstacles. Figure 3 shows the subgoals in the HomeLab (the domestic test environment at Philips Research Eindhoven, The Netherlands), from the robot's current location (L) to a certain goal position (G). The planner outputs the desired change in position in order to reach the first calculated subgoal. By doing so, this ultimately leads the robot to his final goal. The information provided by the localization module is used determine the position of the robot and, thus, to keep track of the change in position already achieved.

The second component of the navigation architecture, the behavioral execution module, receives as input (from the planer) the desired relative displacement for the robot and determines the linear and angular speeds (v and ω, respectively) necessary to perform it. Then, these speeds can be readily transformed to wheel motor commands. The behavioral execution is implemented using a behavior-based control approach. We refer to [2] for an introduction to behavior-based robots. Obstacles which are not in the map, both static and dynamic, possibly show up in front of the robot while moving. To avoid bumping into them, an avoid-obstacle algorithm is implemented. Ultrasonic sensors are used to detect these obstacles.

The cooperation of the fast planner module and the behavioral execution one leads the robot to his goals.

8 High Level Reasoning Module

In order for the robot to realize high level goals it must be capable of reasoning about the information it has about its world. A flexible reasoning mechanism

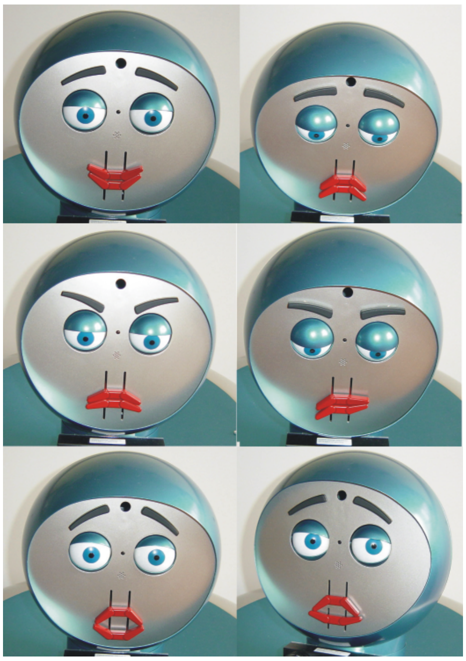

Fig. 4. 3D mechanical head expressing emotions.

that is dedicate to operate in such a practical problem domain as the domestic user environment is essential for a proper functioning of the robot. We plan to use the Belief, Desires and Intention (BDI) architecture that is well known in the field of agent and multi-agent systems. For the experiments we wrote scenario's in a language "Q" and incorporated this in an expert system (CLIPS).

9 Test Results

As far emotion generation is concerned, Figure 4 show some pictures of the head with different facial expressions. Although the actual scientific evaluation still has to be done we have had some first very positive reactions from extensive demonstrations during one week exhibition for totally over 700 people. These demonstrations were conducted by three different relatively inexperienced users (i.e. not the robot developers). By means of some simple dialogues speech recognition, speech synthesis with lip synchronization, emotion generation (facial expressions), object recognition and turn-to-speaker were successfully demonstrated. The general reaction of the observers was appreciation and pleasure.

10 Conclusions

This paper reported the results we have obtained during the on-going development of our domestic user-interface robot. To realize emotional feedback we have built a mechanical 3D head which is controlled by 17 standard RC-servo-motors. The head can express six basic emotional facial expressions. The robot is able to determine the position of the user localizing the origin of any person speaking near him. Additionally, the robot can gather information from the ambient intelligence in which it is assumed to operate and, in the other way around, it can redirect user commands to this environment.

The robot can localize himself in the environment using stereo images and the so-called appearance-based approach. This approach is appealing for its simplicity and due to the stereo vision less sensitive to change in illumination. On the basis of a proper localization, navigation is performed by using two modules: a planner and a behavioral execution module. The planner module calculates subgoal positions for the behavioral execution module in order to prevent getting stuck by obstacles. The Wave Front Distance Field algorithm is used by the planner to calculate the subgoals.

All the modules of our robot are controlled and coordinated in a flexible way using a central controller.

Finally, we presented our software development framework called the Dynamic Module Library. This framework is a state-of-the-art software tool to implement distributed robot applications. An application is runtime configurable by means of a graphical console: the robot application software modules can be probed, started, stopped, removed, added, and connected to each other on-line.

Our project represents a link between two *service to humans* paradigms: service robots and ambient intelligence. Hopefully, other fruitful cooperations would emerge between these two field in the next years.

Acknowledgements

This work has been supported by the European (ITEA) project *"Ambience: Context Aware Environments for Ambient Services"*.

References

1. Aibo, http://www.aibo.com, 2002.
2. R. C. Arkin, "Behaviour Based Robotics", *MIT Press*, 1997.
3. Arras, K.O., Philippsen, R., Tomatis, N., de Battista, M., Schilt, M. and Siegwart, R. (2003) A Navigation Framework for Multiple Mobile Robots and its Application at the Expo.02 Exhibition. in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2003), Taipei, Taiwan.
4. H.J.W. Belt and C.P. Janse, *Audio Processing Arrangement with Multiple Sources*, Patent application PHN 16638 EP-P, 1998.
5. H.J.W. Belt and C.P. Janse, *Time delay Estimation from Impulse Responses*, Patent application PHN 017163.

6. M.E. Bratman, D.J. Israel & M.E. Pollack, Plans and Resource-Bounded Practical Reasoning, *Computational Intelligence*, 4(4):349-355, 1988.
7. A. Bruce, I. Nourbakhsh and R. Simmons, "The Role of Expressiveness and Attention in Human-Robot Interaction" in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation Washington, DC, May 2002*, pp. 4138-4142, 2002.
8. J. Cassell, "Embodied Conversational Agents: Representation and Intelligence in User Interface", *AI magazine*.
9. Elliot, C.D., "The Affective Reasoner: A Process model of emotions in a multi-agent system", Ph.D. Thesis, The Institute for the Learning Sciences, Northwestern University, Evanston, Illinois, 1992.
10. A. J. Davison, M. Montemerlo, J. Pineau, N. Roy, S. Thrun and V. Verma, "Experiences with a Mobile Robotic Guide for the Elderly," in *Proceedings of the AAAI National Conference on Artificial Intelligence* 2002.
11. P. Elinas, J. Hoey, D. Lahey, J. D. Montgomery, D. Murray, S.S. James and J. Little, "Waiting with José, a vision-based mobile robot" in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation Washington, DC, May 2002*, pp. 3698-3705, 2002.
12. Inca plus camera, http: // www.cft.philips.com/industrial vision. 2002.
13. ITEA Ambience project, http://www.extra.research.philips.com/ euprojects/ambience/.
14. B.J.A. Kröse, N. Vlassis, R. Bunschoten and Y. Motomura, "A probabilistic model for appearance-based robot localization". *Image and Vision Computing*, 19(6):381-391, April 2001.
15. O'Reilly,W.S.N., "Believable Social and Emotional Agents", Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.
16. D.A. Reynold and R.C. Rose, "Robust text-independent speaker identification using Gaussian mixture models", IEE Trans. Speech and Audio processing, vol 3, no1, pp 72-83, jan 1995.
17. A. Ortony, A. Clore and G. Collins, *The Cognitive Structure of Emotions*. Cambridge University press, Cambridge, England, 1988.
18. Papero, http: // www.incx.nec.co.jp/ robot/ PaPeRo/ english/ p_index.html. 2002.
19. J.M. Porta, B. Terwijn and B. Kröse, "Efficient Entropy-Based Action Selection for Appearance-Based Robot Localization". In *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, May 2003.
20. Josep M. Porta and Ben Kröse. "Enhancing Appearance-based Robot Localization Using Non-Dense Disparity Maps". In *Proceedings of the International Conference on Robotics and Intelligent Systems (IROS), Las Vegas, USA*, 2003. To appear.
21. S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C.R. Rosenberg, N. Roy, J. Schulte and D. Schulz, "MINERVA: A Tour-Guide Robot that Learns", {*KI*} - *Kunstliche Intelligenz*, pp. 14-26, 1999.
22. J. Vandorpe, "Navigation Techniques for the mobile robot LiAS" *PhD Dissertation, PMA, Katholieke Universiteit Leuven*, 1997.
23. N. Vlassis, B. Terwijn and B. Kröse, "Auxiliary particle filter robot localization from high-dimensional sensor observations". In *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington D.C., May 2002.