

Diligent: Towards a human-friendly navigation system

Rachid Alami*, Igor Belousov†, Sara Fleury, Matthieu Herrb,
Félix Ingrand, Javier Minguez‡, Benoit Morisset
LAAS/CNRS,
7 avenue du Colonel Roche, F-31077 Toulouse Cedex 04, France
{first-name.last-name}@laas.fr

Abstract

This paper presents our project to develop a “Personal robot” called “Diligent”. We first discuss the main issues involved in personal robotics. Then we analyze some of the key features implied by what we call “Human-friendly navigation”. Finally, we present the current state of Diligent. Diligent is already able to navigate safely and repeatedly in a relatively large environment. It is also capable of some preliminary human-robot interactions through very simple and intuitive commands.

1 Introduction

From our point of view, the development of personal robots is a new center of convergence and a motivating challenge in robotics research. One key aspect is “added” to the “standard challenge” of autonomous robots: the essential role of the “human in the loop”. This has numerous consequences. Two of them are of particular interest for us: (1) the robot should be able to operate in an environment which has been essentially designed for human, and (2) the robot will have to interact with human.

The human-centered theme is currently investigated in several areas which are rather different. The spectrum of developments range from humanoids[16] to wearable computing and sensing, human augmentation, telepresence, smart rooms or even intelligent objects[5]. In our case, we aim to contribute to this domain through the “narrow” point of view of robot navigation.

This paper presents our project to develop a “Personal robot” called “Diligent”. We first discuss the main issues involved in personal robotics. Then we analyze some of the key features implied by what we call “Human-friendly navigation”. Finally, we present the current state of Diligent. First of all, Diligent is an open autonomous mobile robot. It is already able to navigate

safely and repeatedly in a relatively large indoor environment. It is also capable of some preliminary human-robot interactions through very simple and intuitive commands. One of our targets is to show that the human interaction capabilities should not hinder its capabilities to navigate. On the contrary, the robot should navigate better. Besides, it should be less disturbing and more compliant to human motions and activities.

2 Human-Friendly Navigation

Diligent’s essential task is navigation¹. Our target, within the framework of personal robotics, is to endow Diligent with “human-friendly navigation”.

This notion has still to be precisely defined. In fact, we envision our project itself as a mean of incrementally defining and investigating the main issues it involves.

However, we can already mention the following issues:

- The ability to operate in an environment open to the public with very little or no adaptation. Indeed, “human-friendly navigation” implies safe (for the human and the robot), robust and reasonably efficient (for the task itself) navigation in such an environment.
- “Human-friendly navigation” means also a mobile robot which is easy to instruct (map and locations learning), easy to use, easy to interact with.
- Another aspect should be the ability of the robot to analyze and react efficiently to human motion around it. For example, it may have to make decisions like “follow”, or “pass”, or “leave room” to a person or a group of persons. In other words, the robot should “comply” with human activities in a given environment.
- One should also consider more subjective issues such as human understanding and human acceptance of the robot behavior.

* Authors list in alphabetical order.

†M.V.Keldysh Institute for Applied Mathematics, Russian Academy of Sciences

‡Centro Politécnico Superior, University of Zaragoza

¹Depending on the context, we will endow it with a set of complementary tasks that will heavily rely on navigation abilities

We envision such a robot as an extension and a new challenge for autonomous mobile robotics.

3 Diligent control architecture



Figure 1: Diligent: a Nomadic XR4000 with LAAS architecture.

Diligent hardware is a standard XR4000 Nomadic robot (Fig. 1) equipped with a SICK laser range finder. Diligent control architecture is based on the LAAS² architecture [1, 2] which has been ported on Linux.

- **The functional level** is composed of a set of “modules” that embed all basic built-in robot action and perception capacities. In the next section, we will present and illustrate the use of the most important ones.
- **The execution control level**, or Executive, controls and coordinates the execution of the functions distributed in the modules, according to the task requirements specified by the decision level.
- **The decision level** which includes the capacities for producing the task plan and supervising its execution, while being at the same time reactive to events from the other levels. It may embed several layers, according to the application; their basic structure is a planner/supervisor pair that enables to integrate deliberation and reaction.

In our point of view, such an architecture is particularly adapted to and should provide a convenient framework for developing human-robot interaction scenarios. Indeed, it is plan-based: this means that the supervisor is built on an explicit reasoning on the tasks and on the robot capacities to achieve them in a given context. Plan-based architecture allows also to exhibit the robot goals and current plans; this can serve as a basis for developing interaction and/or cooperation schemes. We have already successfully experimented such a similar paradigm for inter-robot cooperation ([3]).

Execution Control Level
 Decision Level
 Functions

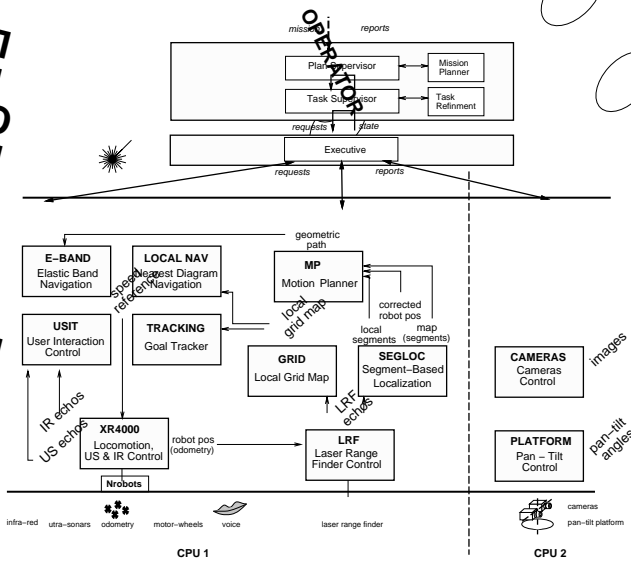


Figure 2: Diligent’s architecture

We briefly recall the main ideas underlying this architecture. It involves three hierarchical levels (Fig. 2) having different temporal constraints and manipulating different data representations.

²LAAS: LAAS’ Architecture for Autonomous Systems.

4 A robust navigation system

We briefly present and illustrate here-below the basic bricks with which we have built the current version of Diligent navigation system: localization, motion planning and motion execution primitives as well as a robot supervisor which controls the overall system.

4.1 Segment-based Localization

The localization method relies on a map of obstacle edges incrementally built by the robot itself from laser range data. Map building and localization procedures use an Extended Kalman Filter to match the local perception with a previously built model [14]). Diligent localization module, called SEGLOC, provides three main localization modes:

- The Re-Location Mode is used when the robot is lost.
- The Continuous Tracking Mode is used when a good probabilistic estimate of the robot position is available. It relies both on robot position variance and on a predicted local aspect.



- The Simple Tracking Mode is used when there is not enough information in the environment to produce a new position (e.g. in a corridor).

Fig. 3 shows a localization experiment in a map of about 400 segments previously built by the robot with the help of a human instructor. The localization process in this relatively large model is performed in less than 120 milli-seconds. We run it at 1 Hertz.

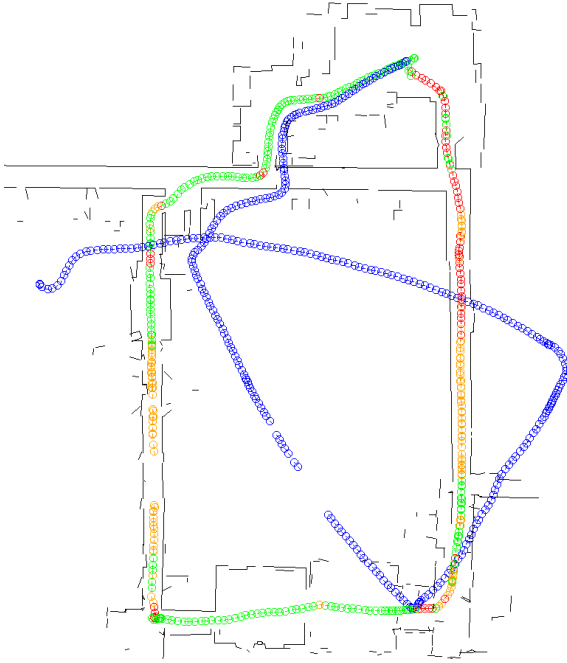


Figure 3: Comparison of position estimation obtained from XR4000 odometry alone (dark circles - missing positions are due to temporarily communication break down) and from the SEGLOC module (grey circles). Dimensions : $45 \times 60 m^2$.

4.2 The Motion Planner

The main purpose of this module is to plan feasible paths for the robot. It is able to compute trajectories for holonomous as well as non-holonomous robots [12]. One of its peculiarities, is that it accepts requests which allow the supervisor to control and to adapt motion planning activities to the current task needs. Indeed, the supervisor can select the level of discretization, the type and the source of obstacles to be taken account by the planner (from models built by the other modules), the shape and the kinematic constraints of the robot. The MP module is used in conjunction with the elastic band (E-BAND module): it produces an initial path which is dynamically adapted during execution.

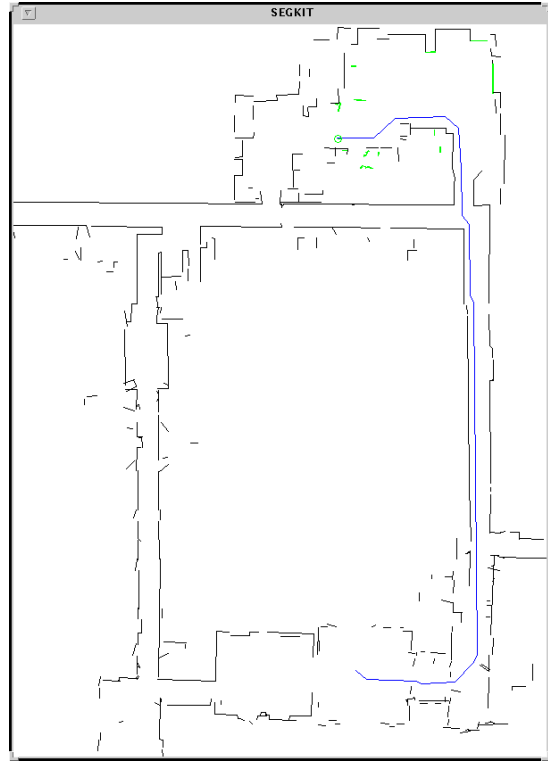


Figure 4: A map of the lab ($45 \times 60 m$) and a path produced by the Motion Planner.

4.3 Elastic Band for Plan-execution

The *elastic band* is a method to dynamically modify a trajectory in order to take into account variations in the obstacle layout between the model used during path planning and the actual sensor data. The method used here was first proposed by S. Quinlan and O. Khatib [15] and further developed and implemented by M. Khatib [10]. The principle is to build a flexible path between the current robot position and the goal, described by a sequence of configurations in free space. Connexity between these configurations is maintained by a set of internal forces that also optimize the global shape of the path. External forces are associated with obstacles and are applied to all configurations in order to maintain the path away from obstacles (Fig. 6).

The computation time of an elastic band is proportional to its length. In order to maintain the real-time capabilities of the robot, the E-BAND module limits the portion of the trajectory that is elastic. If the band retracts beyond a minimum length, new points are added toward the goal along the trajectory computed by MP. Once it has reached the goal, the band retracts until it only contains a single configuration, indicating that the execution is completed.

In order to take into account moving obstacles, if, during execution, the path is blocked the execution of the

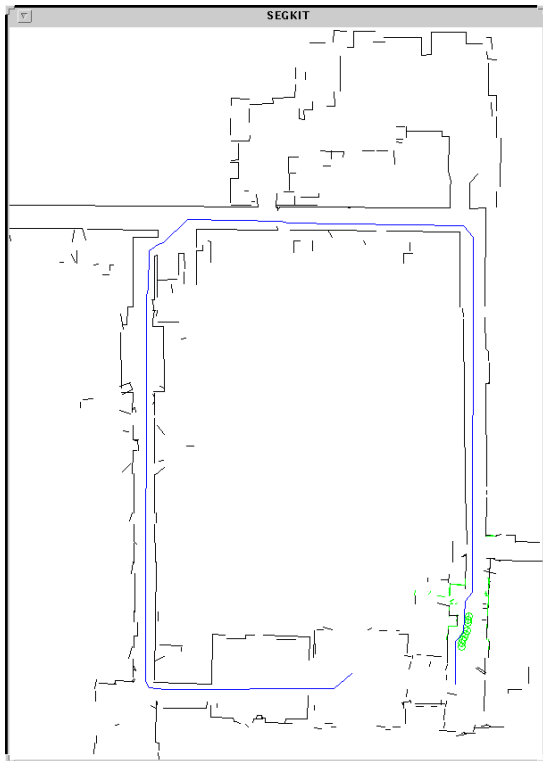


Figure 5: Another path to same goal after having been blocked by a (partially) closed door.

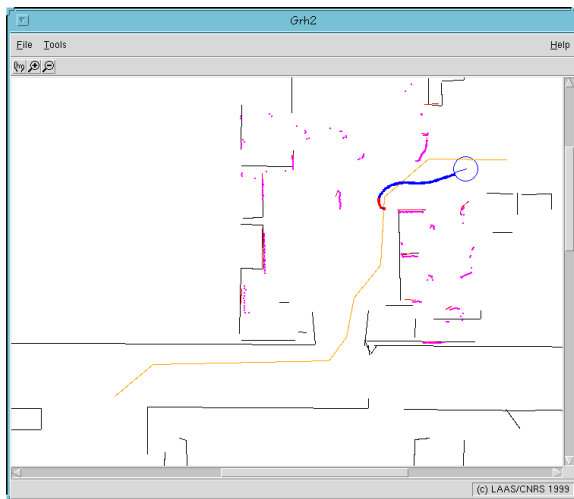


Figure 6: A example of a motion execution based on the elastic band.

band is stopped and a watchdog timer is armed. When the watchdog expires, a failure status is reported to the supervisor. If the next configuration becomes free again before the expiration of the timer (because the obstacle moved), the execution resumes.

4.4 A reactive obstacle avoidance

Diligent is also endowed with a reactive motion execution scheme: the Nearness Diagram method (ND)[9]. ND works in two steps. First, it extracts, from the sensory data, a description of free obstacles regions. It selects one of them (that is usually the closest to the goal location) and evaluates the robot security (based on the distance to obstacles).

For ND, all the possible goal locations and obstacle configurations (relatively to the robot) are described by five general situations. Thus, with the obtained information, ND selects one situation and applies the corresponding law to compute a motion command.

The ND approach is well-suited to deal with constrained and dynamic environments. We intend to use it in a very cluttered environment or in densely populated areas.

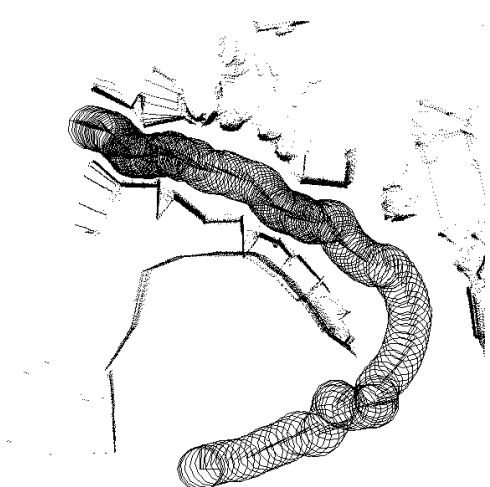


Figure 7: A typical run of ND

4.5 The Robot Supervisor

The robot supervisor is programmed in Propice (formerly PRS) [8, 7]. It provides an interface with all the robot modules thus allowing to program robot control. All supervisor activities (task refinement, control, display, dialog...) are programmed using goal directed and/or situation driven procedures.

The main activities, performed by the Robot Supervisor, in its current version are: Mission Management, Robot Localization Control, Robot Navigation Control, Human/Robot Interaction Control. These different activities interact depending on the current context.

Mission management This activity involves the management of the set of missions that can be incrementally allocated to the robot through several means.

In the current implementation, missions involve essentially navigation tasks. Mission management allows to add or delete missions through textual or through direct human/robot interaction. It is able to postpone, restart or re-plan missions depending on execution contingencies.

Robot Localization Control This activity runs and controls the localization system (see 4.1). It permanently monitors the “quality” of robot localization and implements different levels of reaction depending on the threshold reached by the robot position uncertainty. The first level induces a change in the re-localization modalities. The second level induces an active strategy (e.g. a momentary stop with rotations in order to find landmarks). The third level corresponds to the inability of the robot to solve the problem by itself; it postpones any autonomous motion tasks and asks for human help.

Robot Navigation Control Robot navigation tasks are implemented through the use of a “classical” plan-and-execute paradigm based on a previously learned map. However, this paradigm is enhanced by several features. First, trajectory execution, based on the control of an elastic band, allows an effective robustness to contingencies, from local obstacle avoidance, to adaptation to moving and transient obstacles as well as adaptation to significant re-localization updates. Second, if the band is completely blocked for a period of time, a new plan is searched, taking into account an update of the learned map. The loop is repeated iteratively until the robot reaches its goal or the planner finds no path, or an “external” event entails the postponement or the cancellation of the navigation mission.

Human/Robot Interaction Control This aspect will progressively take an important part in the project. This activity is currently limited at the supervisor level to the control of the function described in the next section.

5 Human-robot interaction

5.1 Various control interfaces

The robot supervisor offers (1) a programmer interface through the Propice programming environment, (2) a textual user interface through a TCP/IP connection and (3) a user and public interface through a compliant motion mode.

Interfaces (2) and (3) allow the user to add or delete navigation missions (e.g. “GO-TO Station-N”). Besides, interface (3) allows the user or any person from the public to set the robot in a compliant mode. The human can then “push” the robot in a compliant but safe (no

collision) manner. The direction of motion is detected using the ring of infra-red sensors while the ultra-sonic sensors are used to create repulsive forces in order to avoid contact with obstacles. This mode is used in two situations: (1) when a human encountering the robot wants to move it aside and (2) to provide to an operator a comfortable and fast mean to guide the robot to a given place for example for allowing a robust re-localization.

Another interface feature is the use of the speech synthesis system to indicate the main changes in robot state as well several displays.

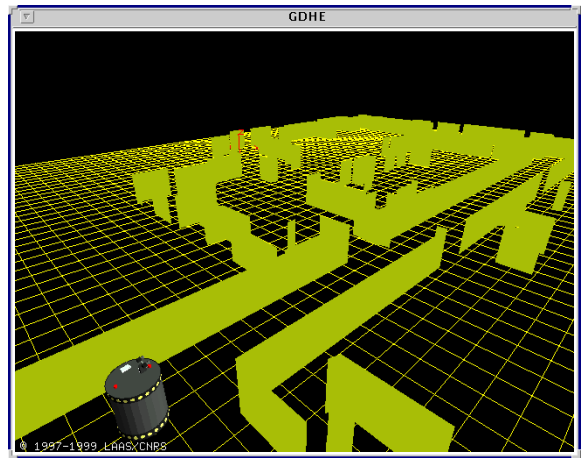


Figure 8: A display of the world as seen by Diligent.

5.2 WEB-based navigation control

In the last few years, many systems for WWW-based robot control have been developed. A list of active systems providing free access through Web browsers is presented on the NASA Telerobotics Web-page[17].

While many systems allow manipulator control in a known environment, there are only a very limited systems that provide mobile robot control. We can cite "Khep On The Web"[11], the Blimp from Berkeley[4] and Xavier[18]. A mobile robot equipped with manipulator arm is presented in [13]. The main characteristic of our system, in its current state, is the ability to control the robot at task level. The operator selects the goal and the robot is entirely autonomous. It plans on-line and adapts its execution to the actual context. However, the operator can request a stop at any moment and can freely control the camera orientation.

Diligent has an embedded HTTP server which provide the applets to the clients. The Java applet connects to a server process on the robot. Through this connection the applet gets the map of the environment and the list of the stations, and it can send orders to the robot (move the pan&tilt platform holding the camera or send

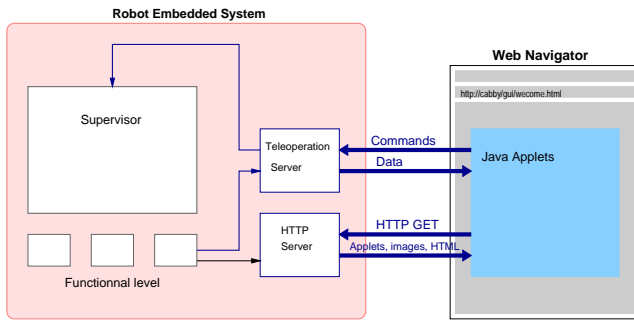


Figure 9: Diligent WEB services

the robot to a station). The orders received from the applet are transmitted to the robot supervisor[7] through a Proprietary message passer.

The image is exported directly to the HTTP server through a cgi script that controls the video acquisition board and sends the compressed images in JPEG format to the Web navigator. The operator may also request a reconstruction in java-3D of the current view of the map[6].

By the time we are writing, the URL is not yet open. We will certainly open it in the future to colleagues and partners. However, it has already been controlled successfully from several places (Paris, Lille, England..).

6 Results and future work

The basic function that we wanted to demonstrate is the ability of the robot to navigate “safely” and repeatedly in the laboratory environment from one place to another. Indeed, a short term goal is to allow its effective use by the lab members, through very simple and very intuitive commands.

We can say that we have succeeded in this first step. Systematic and intensive tests have yet to be performed even though a great number of tests and demonstrations have been performed satisfactorily.

Diligent is capable of long run robust navigation in an environment which can be reliably modeled by horizontal laser scans. The navigation is robust to substantial differences between the actual environment state and the learned map. The various figures presented in this paper illustrate such capabilities. Besides, Diligent demonstrated effective autonomy in planning and executing its navigation tasks, end re-planning in case of permanent obstacle or “non-cooperative” human (we call “non-cooperative” human, a human who blocks completely the path of the robot and who does not leave place after a fixed amount of time - several seconds). Fig. 4 and Fig. 5 show a situation where the robot has been blocked at a door and has decided to re-plan.

However, there is a great difference between running a

robot in a “protected” lab environment and leaving the robot “alone” in an environment open to the public.

Concerning the navigation itself, there is clearly limitation in the localization and obstacle modeling. We will develop a more generic multi-sensor localization in environments populated by human. Besides, it is also necessary to devise a multi-sensor 3D obstacle detection and avoidance allowing to effectively run in almost any-indoor environment. Another key aspect is learning capabilities and ease of programming; indeed, substantial progresses should be devoted to map learning and maintenance.

Concerning the robot supervision and the human-robot capabilities, we will address the following issues.

The robot (the supervisor+various communication means) should be able, whenever it is necessary, to explain its state, its current goal and how it intends to reach it.

We intend to tackle Human-robot interaction in a generic way is to view as a an incremental and interactive problem solving process. Another interesting aspect is to implement a human-robot interaction based on explicit models of the different “types” of humans: programmer, instructor, user, public

These developments will be possible only if we also implement perceptual primitives allowing the detection and localization of human, human motion detection, human recognition, human gesture or movement interpretation.

References

- [1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An Architecture for Autonomy. *International Journal of Robotics Research*, 17(4):315–337, April 1998.
- [2] R. Alami, R. Chatila, S. Fleury, M. Herrb, F. Ingrand, M. Khatib, B. Morisset, P. Moutarlier, and T. Siméon. Around the lab in 40 days... In *IEEE ICRA, San Francisco, (USA)*, 2000.
- [3] R. Alami, S. Fleury, M. Herrb, F. Ingrand, and F. Robert. Multi Robot Cooperation in the Martha Project. *IEEE Robotics and Automation Magazine: Projects funded by the Commission of the European Union*, March 1998.
- [4] The Blimp. *vive.cs.berkeley.edu/ blimp*, currently offline, 1999.
- [5] J. Canny and R. Alami. Workshop on Personal Robotics. In *IEEE ICRA, Detroit, (USA)*, 1999.
- [6] G.J. Clapworthy I. R. Belousov, J.C. Tan. Teleoperation and java3d visualisation of a robot manipulator over the world wide web. In *Proc. of the IEEE Int. Conf. On Information Visualisation IV’99, London, U.K.*, july 1999.

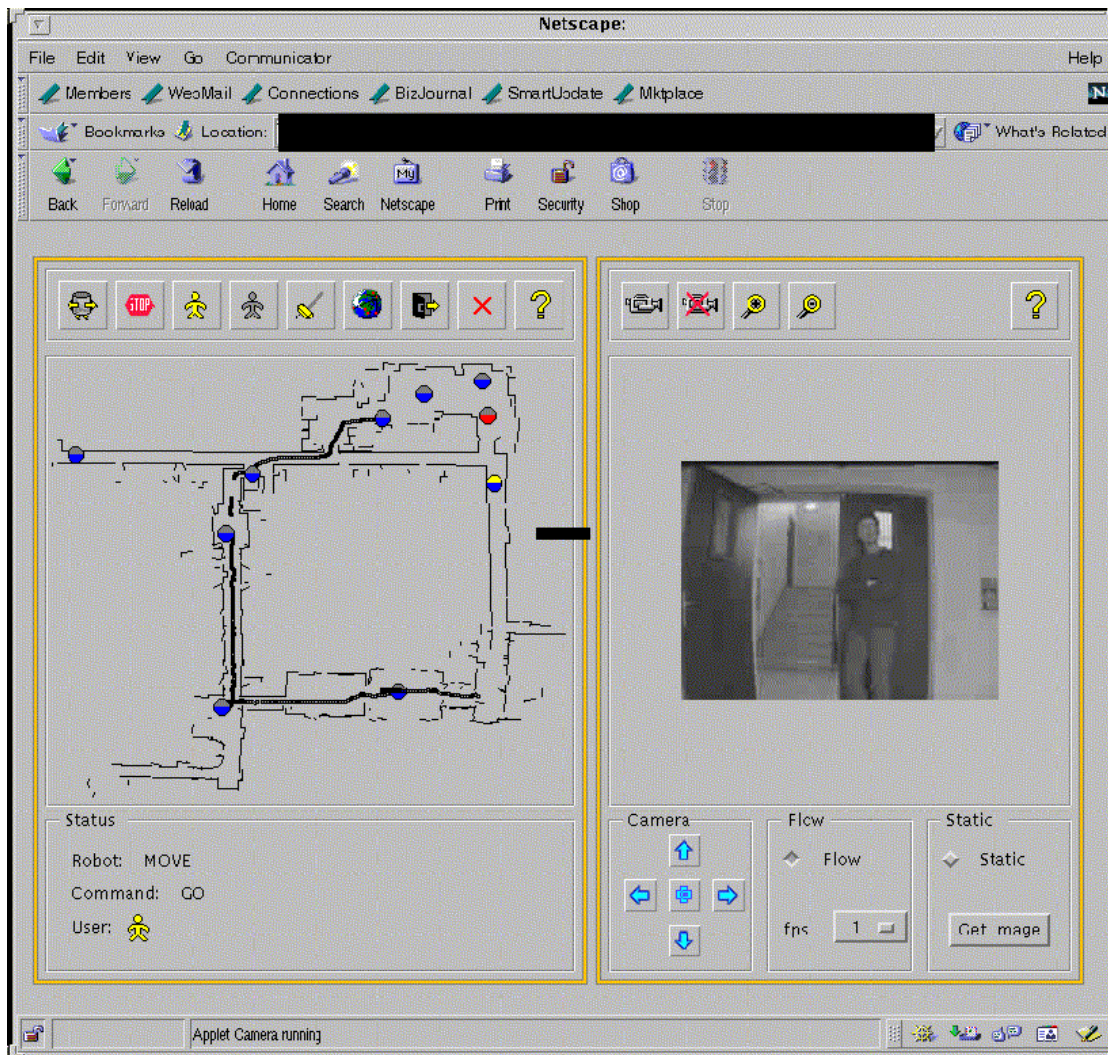


Figure 10: Diligent WEB page

- [7] F. F. Ingrand, R. Chatila, R. Alami, and F. Robert. PRS: A High Level Supervision and Control Language for Autonomous Mobile Robots. In *IEEE ICRA, St Paul, (USA)*, 1996.
- [8] F. F. Ingrand, M. P. Georgeff, and A. S. Rao. An Architecture for Real-Time Reasoning and System Control. *IEEE Expert, Knowledge-Based Diagnosis in Process Engineering*, 7(6):34-44, December 1992.
- [9] L. Montano J. Minguez. Nearness diagram navigation (nd): a new real time collision avoidance approach. In *submitted to IROS'2000 Conference*, 2000.
- [10] M. Khatib. *Contrôle du mouvement d'un robot mobile par retour sensoriel*. PhD thesis, Université Paul Sabatier, Toulouse, December 1996.
- [11] Khepera. <http://KhepOnTheWeb.epfl.ch>, 1999.
- [12] J. P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5):577-593, 1994.
- [13] R. Luo and T. Chen. Remote supervisory control of sensor based mobile robot via internet. In *IROS'97 Conference*, 1997.
- [14] P. Moutarlier and R. G. Chatila. Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modelling. In *Proc. International Symposium on Robotics Research, Tokyo*, 1989.
- [15] S. Quinlan and O. Khatib. Towards real-time execution of motion tasks. In R. Chatila and G. Hirzinger, editors, *Experimental Robotics 2*. Springer Verlag, 1992.
- [16] M. Tanie and G. Giralt (Eds). Europe-japan workshop on human-friendly robotics. In *Tokyo (Japan)*, November 1999.
- [17] NASA Telerobotics Web-page. http://ranier.oact.hq.nasa.gov/telerobotics_page/realrobots.html. NASA, 1999.
- [18] Xavier. www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/Xavier, 1999.