"AGV" MODELLING USING OBJECT ORIENTED TECHNIQUES THROUGH UML LANGUAGE IN FMC

Carlos E. Villanueva Cano
Universidade de Brasilia Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle(GRACO), CEP 70910-900, Brasilia, DF
e-mail cvillanueva@unb.br

Alberto J. Alvares
Universidade de Brasilia Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle(GRACO), CEP 70910-900, Brasilia, DF
alvares@AlvaresTech.com

Sadek A. Alfaro
Universidade de Brasilia Departamento de Engenharia Mecânica e Mecatrônica, Grupo de Automação e Controle(GRACO), CEP 70910-900, Brasilia, DF
sadek@unb.br

***Abstract.**. Object-oriented modelling provides a new point of view to focus a flexible manufacturing system using organized models similar to the Flexible Manufacturing Cell (FMC) presented in this paper which composed by Flexible Manufacture Modules (FMMs) and an automated guided vehicle (AGV). The implemented FMC working with revolution specimens, with predetermined measures length, are composed mainly by the following FMMs: CNC, micrometer, robot, storage system and a mobile robot that works as an AGV. The AGV, as central part in this paper, is also composed by ultrasonic, infrared and tactile sensors, which, through properly implemented and organized algorithms permit an optimum navigation. The mobile robot navigation is based mainly by a Vision System which consists in the CCD camera recognition of lines and landmarks properly arranged in the FMC. The supervision and material handling are the main function of AGV inside of the FMC. In order to the mobile robot acquire a similar behaviour of a Material-Handling System should develop operation characteristics similar to it (capacity, load, discharge, point of load, point of discharge, busy time, etc.), behaviours that relates navigation functions (recognition by vision and sensors, selection of velocities, times, etc) and AGV functions (position, direction, turns, etc). These models can be used to achieve an effective organization and control of the AGV as part of the FMC and to reach the objectives that emphasize the word flexibility in the field of the manufacture.*
***Keywords**: Automatic Guided Vehicle (AGV), Flexible Manufacturing Cell(FMC),Undifined Modelin Language(UML), Nomad XR4000, Vision tracking, Navigation.*

1. Introduction

Nowadays robot mobility, working specifically as Automatic Guided Vehicle - AGV, is necessary in modern industrial environments for a number of functions, such as transportation of workspace, material handling or performance of specific task at different stations.

The most utilized word for today's manufacturer is "agility", that it a most important characteristic of the flexibility manufacturing. An agility manufacturer is one who is the fastest to the market, operates with the lowest total cost and a lowest time of speed of delivery. AGV as part of a FMS or more specifically, in our study case, in a FMC is direct related with the terms that characterize a FMC, "agility" and flexibility. The supervision and material handling are the main function of AGV inside of the FMC. In order to the mobile robot acquire a similar behavior of a Material-Handling System should develop operation characteristics similar to it (capacity, load, discharge, point of load, point of discharge, busy time, etc.), behaviors that relates navigation functions (recognition by vision and sensors, selection of velocities, times, etc) and AGV functions (position, direction, turns, etc).

The objective of this paper is to describe the AGV modelling using object oriented techniques through UML language, more specifically a vision navigation technique based in line tracking that will be implemented for achieved the own functions as a "Material Handling".

The first part of this paper describes the modelling working environment where the Nomad XR4000 will work as an AGV. It will be presented the disposition of the line and landmarks as well as the machines that will be working in the flexible manufacturing cell. Nomad's software and hardware systems will be presented briefly in the first part. The software will be treated more extensively in the chapter six for defining the classes involved in the AGV modeling. In the modeling part using object oriented techniques will present three diagrams specified by UML language such as: use case diagram, class diagram and sequence diagram.

This work is a part of the project WebMachining (http://WebMachining.AlvaresTech.com) that describes a proposal of methodology for CAD/CAPP/CAM integration in remote manufacture of rotational pieces utilizing to Internet in special the protocols associated to the World Wide Web.

2. Modelling Working Environment

The environment where the mobile robot Nomad XR4000 will work as a "Material Handling"(Booch, A.,1998) in a flexible manufacturing cell is composed mainly by the following FMMs: CNC Center Turning (Romi Galaxy 15M), micrometer laser (Mitutoyo), industrial robot (ASEA IRB 6/2), storage system (Pallet) and a mobile robot (Nomad

XR4000) that works as an AGV. Figure 1a) shows the FMC units that are being implemented in the laboratory as well as the differents kinds of communications that involved.

Figure 1b) shows the landmarks that are represented the main points that the robot should recognize for its localization. These points are point 1 (start functions point) point 2 (zone of discharge of the piece manufactured), point 3 (middle point of nomad navigation) and point 4 (zone of charge of the piece manufactured) (Beccari, 1997). The landmarks utilized in this work were selected as the best landmarks for its use in tasks such as a precise positioning of AGVs or mobile robots in front of loading / unloading (Amat, 2001).

For monitoring the FMC have been implemented the following link http://video,graco.unb.br. The link http://webcam.graco.unb.br has been developed for monitoring the Nomad garage and http://Nomad.graco.unb.br link has been developed for achieving an Nomad upper teleoperation via internet.

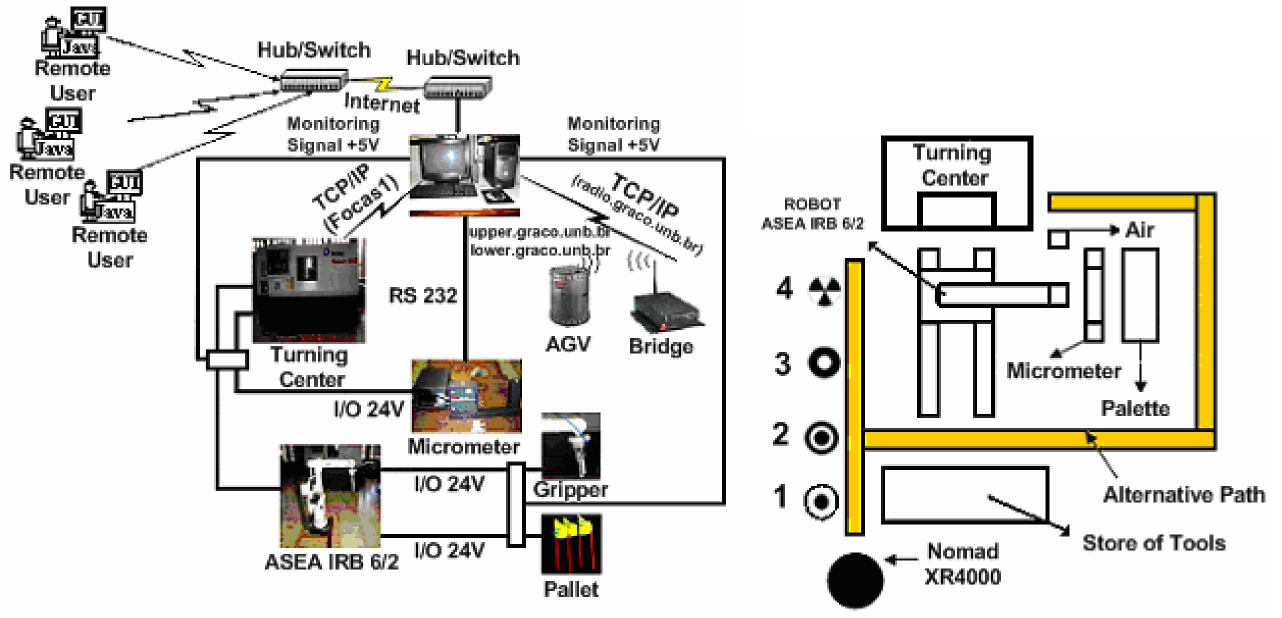

Figure 1. a) Flexible Manufacturing Cell in implementation b) AGV working environment

3. The Nomad XR4000 Mobile Robot

The Nomad XR4000 mobile robot is an advanced mobile robot system that incorporates power management networking, a holonomic drive system, sensing, communication and software development technologies. It can be equipment with other additional features such as a lift mechanism or arm robot.

The Nomad XR4000 mobile robot, also is an integrated system designed for industrial applications or research, Its features include (Nomadic Technologies, 1999): Onboard dead reckoning system (Odometer), Infrared, sonar and tactile sensors, Two PC Pentium computers running the Linux operational system (programmed in C language) interconnected by TCP/IP protocol, Camera, pan-tilt unit and frame grabber for vision task, and Wireless network connection.

The XR C8 Holonomic Drive System incorporated on Nomad XR4000 offers three full degrees of freedom (x, y, ɵ) without restriction in ground clearance, vibration or mechanical complexity. Furthermore, this system is accomplished by employing caster wheels having independently powered steering and translation axes. The XR C8 uses four wheels, resulting in an eight axis, under constrained systems. To control such a system, the Nomad XR4000 uses a specialized motor controller with three DSP's (Digital Signal Processing) and a dedicated 32-bit microcontroller to control all eight axes while estimating the dead-reckoned position.

The Nomad XR4000 has three standard sensor systems: tactile, ultrasonic, and infrared. The XR also has many available sensor options including monochrome or color vision, laser and compass systems.

In reference to the software features, the Nomad XR4000 operating system is Linux, and the daemon Nrobot, it is the software which is "talking" with the hardware. The Nomad has a library (Nhost_client.a) for working with it and giving the opportunity to access the structures, using the sensors, moving, etc. This library is linked with the user program and it is the one who is "talking" with the daemon

4. Nomad XR-4000 Functions in the FMC

The XR4000 main functions in the FMC are: a) Robot navigation through line recognition and landmarks drawn in the factory floor, defining a trajectory of the Nomad as well as its localization during the navigation; b) Storing finished, reprocessing and default pieces, through of a "pallet" that it is placed on top of the robot divided in three spaces for each kind of piece with weight limitation problem; c) "Material Handling", pick up and transporting tools during the "setup" the machine and its maintenance and d) The Nomad XR4000 interact directly with the manager cell.

4.1 Interaction Nomad XR-4000 Functions with the Manager of the FMC

All position and localization points of Nomad XR4000 in the FMC will be controlled for the manager cell. These positions and localizations points are depended of Nomad XR4000 functions that are carried out in the FMC. These functions are: a) Nomad interaction functions working as a material handling during the manufacturing process in the FMC and b) Nomad interaction functions working as a material handling during the setup or maintenance of CNC machine.

4.2 Nomad interaction functions working as a material handling during the manufacturing process in the FMC

The manager cell will send tasks as orders that they will be executed for the Nomad XR4000. These orders are basically robot position and localization. These position and localization points that they will be drawn in the ground are: a) The exit point of nomad for start its functions (point 1) in the FMC, b) Position and localization of nomad in the zone of shipment (point 4), c) Position and localization of nomad in the zone of finished piece discharge (point 2) and d) Position and localization of nomad in the middle of the point 2 and 4.

Nomad will have others lines as alternatively trajectories that they will be implemented in the ground of the FMC. The election of the trajectory will be controlled through position and localization points and additionally sensor fusion that they indicated which trajectory the nomad will have to follow (see Figure 1b).).

4.3 Nomad interaction functions working as a material handling during the setup or maintenance of CNC machine.

The main interaction functions of Nomad XR4000 working as a material handling during the setup or maintenance of CNC machine are: a) Change of the functions during the manufacturing process for the material handling functions during the setup of the CNC machine. The start point after the change of functions will be the point 2. Here an operator will put the tools on the Nomad robot for carrying them near to CNC machine (point 4), b) Start functions, c) Positioning nomad in point 2 for carrying tools, d) Positioning nomad in point 4 for delivering tools and d) When the manager cell stop nomad functions the robot will be positioned in the point 1 after that it will receive other task or the functions will be finished.

5. Robotic Navigation

The navigation is a main problem of any mobile robot and consists of the automatic movement of one place to another through of a coordination of the path planning, the sensor system and the control system. The main objective in all navigation is arriving at pre establishing point without collisions with others knowing or not objects. The navigation can be divided in three sub-task: mapping and modelling of the environment, planning and selection of the path (path planning and selection), and avoid collisions and obstacles (collision avoidance). The function navigation hierarchy of the relationship between these tasks is presented in the figure 2 (Tourino, 2002).

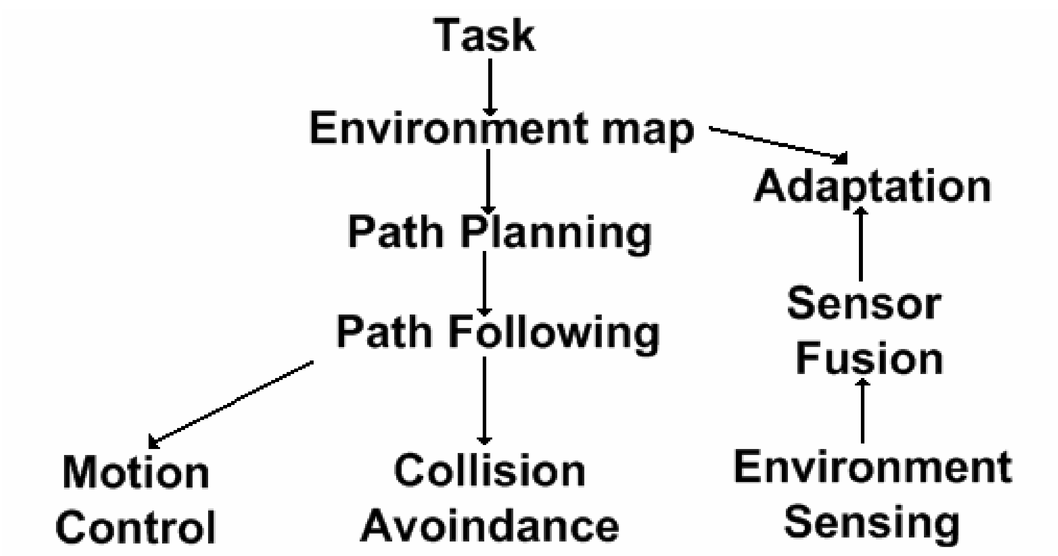

Figure 2. Hierarchy of navigation functions

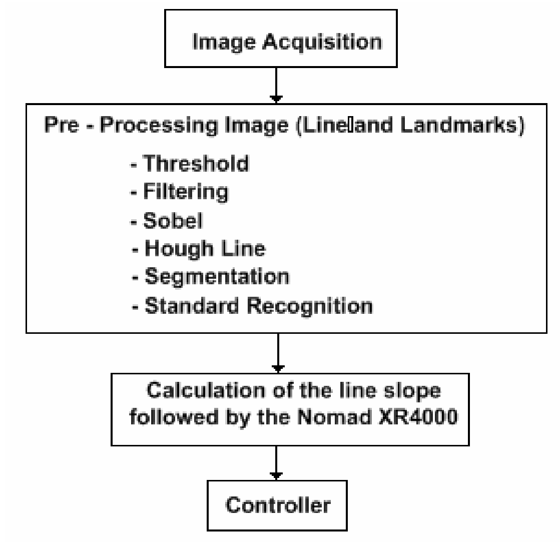

Figure 3. Diagram of lines and landmarks image processing utilized in the XR4000 Nomad navigation

a)

b)

Figure 4. a) Captured Image; b) Image after preprocessing for line detection.

The developed architecture will show a kind of navigation that consist in continuing a line designed in the ground and the recognition of lines and "landmarks" as well as detection of obstacles, that it could be detected during the operation of the FMC. This navigation will be carry out through of the image processing system and fusion sensor (infrared, ultrasound and tactile sensors) of the mobile XR4000 Nomad robot.
The main aspects to achieve this architecture are: a) Acquisition and processing of sensor data of infrared sensors, b) Acquisition and processing of sensor data of ultrasonic sensors, c) Acquisition and processing of sensor data of tactile sensors, d) Acquisition and processing of CCD image captured camera and e) Action, selection and execution.

Aspects a), b) and c) in the implemented system are utilized for avoiding obstacles in robot navigation, aspect d) is utilized for developing a vision based-line tracking navigation system and the last aspect is utilized for the movement control of mobile robot.

Figure 3 presents a general sequence diagram for recognition of lines and landmarks. Included main aspects are: image captured for CCD camera, pre-processing image working with robot image processing system and control movements and dead recognition that works with robot odometer system. Also the pre-processing aspect will be implemented with algorithms such as, threshold, filtering, canny edge detector, Sobel, Hough line, segmentation and standard recognition.

5.1 Line and Landmarks detection

The main visual operation required for AGV (Nomad XR4000) for achieved its functions of navigation is to compute the direction of the track on the floor. This computation estimates the slope of the track as linearized through a pair of its points in the image (Beccari, 1997). The main steps for achieving the detection line are: a) image acquisition showed in Figure 4a, b) filter Image captured with a linear filter (Sobel) with a appropriate threshold for line detection showed in Figure 4b (Trucco, 1998) and c) Applying algorithm for line slope calculation.

For landmark recognition, besides continuing the same steps indicated for lines detection, two additionally steps are necessary: segmentation the preprocessing image for detect the desired symbol and its standard recognition.

6. UML Diagrams

AGV modelling will be working with Rational Rose software, that it is an object-oriented UML software design tool intended for visual modeling and component construction of enterprise-level software applications. UML diagrams commonly created in visual modeling tools include: Use Case Diagram, Class Diagram, Interaction Diagrams (Sequence Diagram and Collaboration Diagram,) State Diagram, Activity Diagram, and Physical Diagrams (Component Diagram, Deployment Diagram).In this paper will be presented three diagrams specified by UML language such as, use case diagram, class diagram and sequence diagram.

6.1 Use Case Diagram

The Use Case Diagram is a graphic view of some or all actuators, utilization commands and their interactions, identified for a system (Quatrany 2000). The utilization commands in the studied case are: Sending order to initialize the functions of the nomad, Selecting the tasks to be executed by the Nomad (see section 4.1), getting data of Nomad

XR4000 when the task was concluded, and Sending order to stop the functions of the Nomad. The figure 5 presents the use case diagram for AGV modeling.

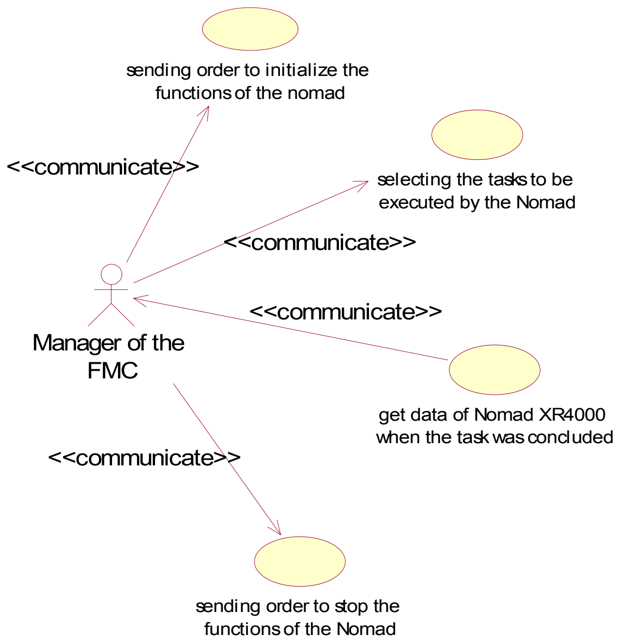

Figure 5. Use Case Diagram

6.2 Class Diagrams (Okamoto 2004, Nomadic Technologies 1999)

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects (Quatrany 2000). The figure 6 shows the UML class diagram for modelling AGV working in a FMC.

In this paper the main object is modelling a navigation technique of AGV working in the FMC with structured environments. This navigation use the vision system as the main sensorial system of the Nomad XR4000 for achieved the goals in the FMC. Infrared, sonar and tactile sensor (fusion sensorial) will work for avoid possible obstacles that they can present when the FMC is working.

Task Class:

This class interacts directly with the manager cell and is responsible for specifying the task to be performed by the Nomad XR4000. This task will be achieved through the robot navigation and using a robot and hierarchic approach, the Task class will make use of one object of the Robot class only. This class has as main attribute a goal, represented for example, as recognition of landmark of the position desired. The main methods of this class are to select a goal (*SetGoa*l) and to verify if the task was completed (*GoalAchieved*).

Robot Class:

This class is responsible for describing the robot that will be used in the task. The robot class has two subclasses with composed relationships called aggregation class. These classes are Actuator Class and Sensor Class. This class has as a main attribute a *NRobot*, that is the server that communicates with the robot hardware by sending commands and receiving sensor data. All user programs interact with robot through this process. The main methods of this class are *N_RobotState* for sending sensor data and *N_GetRobotState* for getting sensor data.

Sensor Class:

This class is responsible for controlled all sensorial data originating from many sensors included in the Nomad XR4000 such as Bi-level Tactile System (Sensus 150), The Ultrasonic Ranging System (Sensus 250), Infrared Proximity System (Sensus 350), Vision System and Integrated Configuration (dead-reckoned position).

Actuator Class:

This class is responsible for the movement of the robot to the position desired. The XR4000 has three convenient and intuitive axes of motion: X, Y and rotation. It also has can accelerate in any direction at any time, making it holonomic. The main attribute of this class *are N_Axix* and *N_AxisSet* structure which contains all the information concerning the base motion axes. The main methods are *N_SetAxes* and *N_GetAxes*.

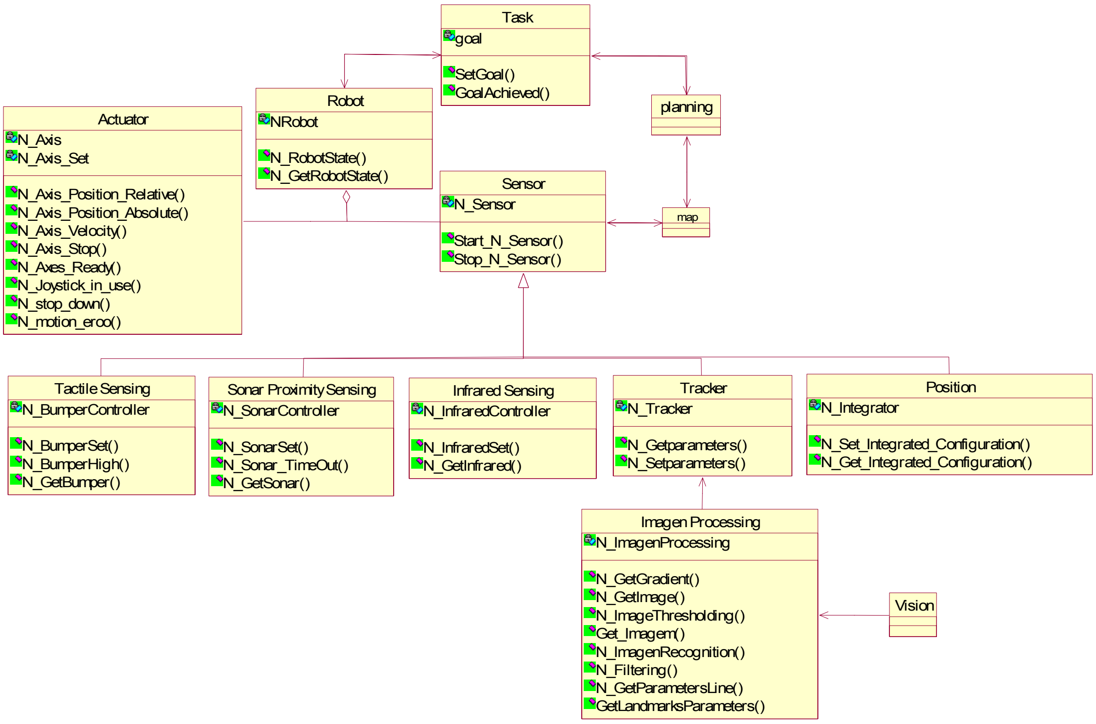

Figure 6. Class Diagram.

Tactile Sensing Class:

This subclass is responsible to provide 100% tactile coverage over the vertical surfaces of the Nomad XR4000. There are 48 bi-level sensing elements that surround the top and bottom perimeters of the Nomad XR and provide both the exact location of contact as well as information about contact force. Additionally, each door has four "floating" switches that register when any part of the door makes contact with an obstacle.

The main attribute of this class is *N_BumperController* for controller all sensorial tactile data registered in the environment. The main methods are *N_BumperSet, NBumperHigh*, for hard hits and *N_GetBumper*.

Infrared Sensing Class:

This class is responsible to provide range information to nearby objects (tipically less than 30 to 50 centimeters away). They determine range by emitting infrared energy using high-current LED's and sensing the amount of returned energy with infrared photodiodes. The returned energy is inversely proportional to the distance to the closely object – thus, these sensors are used as distance or proximity sensors. The returned energy is also a function of the objects reflectivity. The main attribute of this class is *N_InfraredController* for controller all sensorial infrared data registered in the environment. The main methods are *N_InfraredSet* and *N_GetInfrared*.

Sonar Sensing Class:

This class is responsible to provide range information to objects that are relatively far away (between 15 and 700 centimeters). Distance information is obtained by multiplying the speed of sound by the "time of flight" of a short ultrasonic pulse traveling to and from a nearby object. The sonar controller structure holds all configuration data that is valid for all the sonar sets on the robot. Sonar sets are groups of sonar that act together. The main attribute of this class is *N_SonarController* for controlling all sensorial data registered in the environment. The main methods are *N_SonarSet* for selecting required parameters of sensor, *N_SonarTimeOut* and *N_GetSonar* for acquiring sensor data during the navigation of mobile robot.

Position Class:

This class is responsible for the constantly estimation of the Cartesian position (x, y, rotation) with respect to a global coordinate frame that is "draw on the floor". This is often called the dead-reckoned position, and it is useful as an "extra sensor" that indicates where the robot is located in the environment. It is estimated by measuring changes in position (dX, dY, dRotation) over very small time increments (typically 5s) and integrating those changes over time. Hence it is called the Integrated Configuration. The main attribute of this class is *N_Integrator* that it interacts with *N_RobotState* and controlled all sensorial data registered. The main methods for this class are, *N_GetIntegratorConfiguration*. Through this method the integrated configuration can be obtained. The integrated configuration can be set to any configuration by modifying the contents of the *N_Integrator* and making a call to *N_SetIntegratedConfiguration*.

Tracking Class:

This class is responsible to describe visual tracking sensor. Through this sensor is possible to track lines and landmarks as object in the image and to retrieve some information about the visual changes suffered by this object. This kind of sensor utilized the vision system of Nomad XR4000. Robot XR4000 has a Sensus 460 system composed for a color composite camera standard NTSC (National Television Standards Committee) and PCI (Peripheral Component Interconnect) frame grabber card with 45 Mbytes/s of transfer data. This system is helped for a pant-tilt unit that is controlled for a serial port RS-232 for the movement of the camera. In Nomad XR4000 the tilt degree of freedom of the Pan-Tilt Unit is held at the maximum downward orientation of about 45^o when looking for the line (Beccari, 1997). The main function of this class is to control the movements of the robot. This control can be achieved through two forms: a) Feedback control along the lines, working witch the slopes of the lines computed for the Image Processing Class through a pair of its points in the image and the position of the robot, and b) Using Kalman filter for predicting the position of the robot at any instant on the basis of its previous positions. The main attribute of this class is *N_Tracker* for controlled all sensorial data acquired for the subclasses Image Processing, Position and Vision Class. The main methods are *N_GetParameters* and *N_SetParameters*.

Image Processing Class:

Image processing is the class responsible for the pre-processing and movement extraction of the sequence of images. Movement extraction is achieved through running processing image algorithm who it gives as results numerical data that they are utilized like entrances for the control system. Image processing for AGV navigation involves low level enhancement line detection and computation of line orientation, identification of crossing among lines and landmarks detection and recognition. Line detection unless an intersection is approaching, the main visual operation required for AGV navigation is to compute the direction of the track on the floor. This computation estimates the slope of the track as linearized through a pair of its points in the image. Nomad XR4000 also relies on visual perception to identify landmarks which suggest or constrain robot behavior and navigation. Line and landmarks were showed in Figure 1. The main attribute of this class is *N_ImageProcessing* who is controlled all process of the sequence images captured until get line orientation or landmark detection. The main methods are *N_GetGradient, N_GetImage*, *N_ImageThresholding, N_Filtering, N_ImagenRecognition N_GetParametersLine and GetLandmarksParameters*.

Vision Class:

This class is responsible for the capture of images. Nomad XR4000 has like a vision system a board of capture video Matrox Meteor and a color camera Hitachi KP-D50. The board of capture permits to get images until 30fps (frames per second) and image resolutions of 640 x 480 for images of 24 bits. The configuration of the board of capture is made through the library ioct1_meteor.h, installed in the Nomad, defines the structures of the data and variables for this goal. The camera has lens of 6mm, f/1,2 with focus and diaphragm manual, without zoom. This class works directly with the Tracking, Image Processing and Position class in the control of the movements of the robot.

Planning Class:

Planning is the class responsible for the path planning algorithms, for example “The Bug Algorithm”, “Alg2” Algorithms that could be implemented for avoid any no predict obstacle. The planning class uses the map class to take decisions about the path the robot should follow. The main attribute of this class is the goal position set by the task class. The main methods of this class are: select the goal, find the path selected and get the path selected.

Map Class:

This class is responsible for represent the environment that surrounds the robot and management the different class of maps utilized in navigation task selected. Maps are represented utilizing occupancy grids and indicate the current robot position.

6.3 Sequence Diagrams

Sequence diagrams demonstrate the behavior of objects in a use case by describing the objects and the messages they pass. the diagrams are read left to right and descending (Quatrany, 2000).

The sequence diagram shown in the figure 7 is specifically about the vision based-line tracking navigation that the robot should continue for achieved its task as a “Material Handling”. This sequence starts selecting, for the manager cell, a task that will be executed for the Nomad XR4000, and then the task class will send the goal that the class robot must achieve interacting with the hardware of the XR4000 that they are represented for the actuator class and the sensor class. The class robot, after receiving the start function initializes the class functions, tracker (image processing and vision class), position, actuator and sensor (infrared, sonar and tactile sensors). The task class will send the goal to be achieved for the tracking class, after that this class will sent a request to the robot class for reading the position where the robot XR4000 is localized. The robot class will request to the class position the coordinates of the position of robot, them it will send the position required by the robot class. The class robot will send the position to the class tracker who will compare with the class task if the goal was reached. If the parameters required by the class task are correct, the robot will stop to be moved but if they aren’t the class task will send an order to continue the process until be achieved the goal. Once that goal has been achieved the class robot will send an order to the class actuator to stop the movement of the robot.

7. Conclusions

This paper described an “AGV” modelling using object oriented techniques through UML language in a flexible manufacturing cell, using a vision-based guidance scheme. Visual system, as a most important sensor for this navigation, is exploited for both line tracking and landmarks recognition. Consequently, providing the basis for an easily reconfigurable vehicle guidance system. This kind of navigation could become very versatile, because the robot could learn environment topology during a training phase and then determine appropriate routes amount specific locations, or it could modify its route during execution based upon the specific set of landmarks perceived.

An UML model of the classes and some sequence diagrams were presented showing how the classes can interact with each other for modelling AGV using vision system working in a flexible manufacturing cell (FMC).

Vision-based line tracking and navigation is developed and implemented in structured environments and the cost for achieved this navigation is low.

Mobile Robot Nomad XR4000 is equipped with software and hardware systems required to work as an Automatic Guided Vehicle (AGV) in flexible manufacturing process.

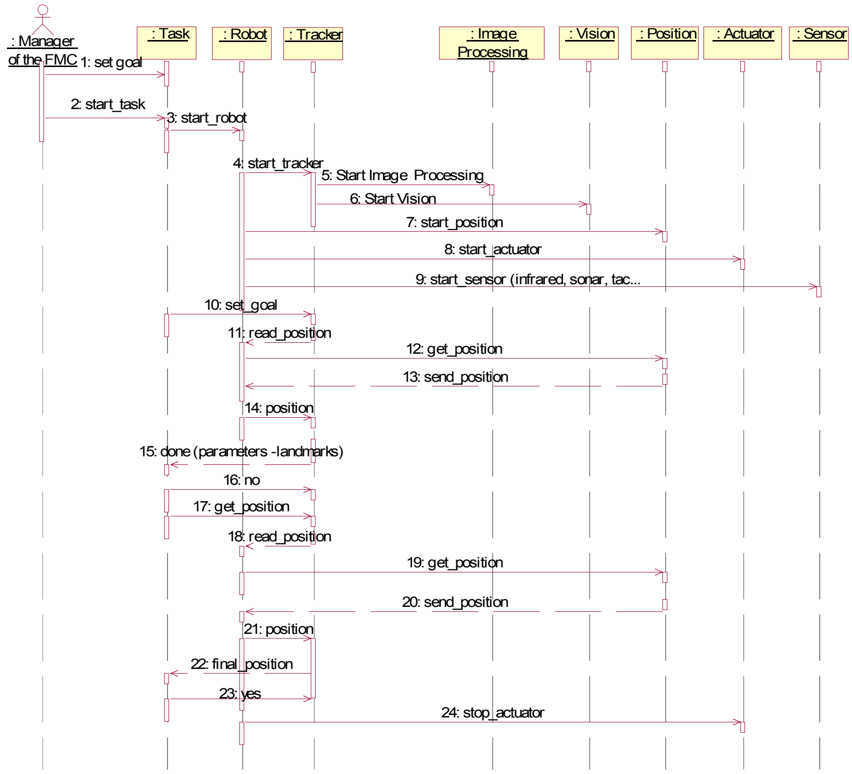

Figure 7. Sequence Diagram of Vision Navigation

8. References

Álvares, A. J., Andriolli, G. F., Dutra, P.R.C., M. De Sousa M., 2003, “A Navigation and Path Planning System for the Nomad XR4000 Mobile Robot with Remote web Monitoring”, Sao Paulo, COBEM2003.

Amat, J., Aranda, J., Casals, A., Fernández, X., 2001, “Optimal Landmark for Precise Mobile Robots Dead-Reckoning”, IEEE International Conference on Robotics and Automation, Seoul, Korea.

Booch , A., 1998, “Object – Oriented Modeling for Flexible Manufacturing Systems”, The International Journal of Flexible Manufacturing Systems, Boston.

Booch, G., Jacobson, I., Rumbaugh, J., 1998, “The Unified Modeling Language User Guide”, Addison-Wesley Pub.Co.

G. Beccari, S. Casseli, F.Zanichelli, A.Calafiore, 1997 “Vision-based Line Tracking and Navigation in Structured Environments” Dipartimento di Ingegneria dell’Informazione University of Parma – Italy

Nomadic Technologies, “Nomad XR4000 Hardware Manual Release 1.0”, California, USA, 1999.

Okamoto J.,J, Grassi, J, V.,Correa, F., 2004, “Modeling autonomous Mobile Robot System with an Object Oriented Approach”, ABCM Symposium Series in Mechatronics – Vol. 1 – pp.25-32.

Quatrani, T., Booch, G., 2000, “Visula Modeling with Rational Rose 2000 and UML”, Addison Wesley Longman.

Tourino, S., 2002, “Sistema de Rastreamento para Robôs Móveis Utilizando Visão Embarcada”, Universidade de Brasília, Brasília DF.

Trucco, E., Verri, A., 1998, “Introductory Techniques for 3-D Computer Vision”, Prentice Hall, USA.