PK Programmierpraktikum II
SS 1999

Robot Construction Kit

Version 0.1

Team 1
ECKSCHLAGER Manfred (mekschl@cosy.sbg.ac.at)
GANGL Thomas (tgangl@cosy.sbg.ac.at)
HUSKIC Zenon (zhuskic@cosy.sbg.ac.at)
WOLACEWICZ Mateusz (mwola@cosy.sbg.ac.at)

Coordinator
Mr. SCHWAIGER Roland

March 17, 1999

Abstract

This paper presents planning and implementation of Robot Construction Kit (RoCK), that enables the user to construct different types of agents(robots) in 3D environment. Implementation should be done with the means of Java and Java3D.

Keywords: agent (robot), 3D environment, construction.

Contents

Chapter 1

Introduction

As a part of our education here at the University of Salzburg we are obliged to take part in a project and implement the same. The project should be implemented through the team effort, as well as with help of our coordinator, Mr. SCHWAIGER, Roland. One of the main reasons why we are implementig this product is DiMAS.

1.1 What is DiMAS?

DiMAS stands for Distributet Multi Agent Simulator, and it is simulation software for representation of several agents(robots) in predefined environment. DiMAS resulted from Number5, which was developed during the last few years by several people under the supervision of HUBER, Reinhold. It was written in Java, and for the time being it is enabled only for 2D representation.

1.2 What is robotics?

This section serves to define the words robot and robotics, as well as to explain where did the word 'robot' come from.

1.2.1 Definition of robot

The word 'robot' was first pronounced by Karel Capek, the Czech playwright, and it comes from the Czech word for forced labor or serf. There are few

definitions of a robot, and here are some of them:
"A reprogammable, multifunctional manipulator designed to move material, parts, tools, or specialised devices through various programmed motions for the performance of a vaiety of tasks."
Robot Institute of America, 1979
Here is another one:
Force through intelligence. Where AI (Artificiall Inteligence) meet the real world.
According to the Webester's dictionary: *An automatic device that performs functions normally ascribed to humans or a maschine in the form of a human.*
The term 'robotics' refers to the study and use of robots. The word was first used by Russian-born American scientist and writer Isaac Asimov, who wrote many books of science fiction.
One definition of robotics is [1]:
Robotics is the intelligent connection of perception to action.

1.2.2 Robot classification

A robot can be decomposed into a set of functional subsystems: mechanical, electrical, sensor, software, process, planning and control.
According to [1] robots have been classified according to their generation, their intelligence level, their level of control, and their programming language level. The generations are:

- Playback robots
- Sensor-controlled robots
- Vision-controlled robots
- Adaptively controlled robots
- Artificially intelligent robots

1.2.3 Sensors

Sensors are grouped into two classes: internal and external sensors[1]. Internal sensors measure variables within the robot - e.g. joint angle, wrist force and platform velocity.

External sensors measure the environment - e.g. range, vision, and voice. Sensor installation poses a number of problems for the robot designer. Often sensors have to be mounted in awkward positions, such as on the fingers of a gripper, where they have to be protected from impact, dust, and moisture. In robotics, sensors are used for:

- measuring robot parameters for control loops
- finding the location of objects
- correcting for errors in the robot's models of itself and of the world
- detecting and avoiding failure situations
- monitoring interaction with the environment such as forces during compilant motion
- monitoring the environment for changes (such as temperature) that may affect the task
- inspection the results of processes

1.3 Requirements

Our task is to write a program for designing agents(robots), and it should be done in 3D representation by the means of Java and Java3D. The following are the most important requirements, that were given by Gerhard Katstaller, who personally participated in DiMAS development (the first considerations):

- Construction of an agent (that is a Robot) should be inplemented by using interactive 3D interface. Physical properties should be enabled for editing. (for example: sensitivity of a light-sensor).
- Regarding light and sound (infra-red); it should be checked how is it solved in Java3D.
- The file format should comply with the KIF (Knowledge...).
- Construction Kit should be applicable for other elements as well, just like obstacles, light, lamp, sound source, etc.

- A Robot, that was constructed by the means of the construction kit should be available for modifications later on as well. This means that save function should be enabled to save all changes in 3D environment. Load function should be able to restore the 3D-Graphics. Import function should be available for DiMAS, too.

- An improvement of DiMAS could be reached if a reasonable Format is produced (it is reasonable due to Java3D).

- The following basic elements could be used for construction: cube, rectangle, circle, cylinder, pyramid, cone, sphere, wheel...

- What about the movable parts just like: wheels, robot arms, etc? Attributes: measurment, position, colour, transparency, friction?

- Sensors: light, infra-red, heat, (what is hearer?)

- Actors: Light(colour), sound, heat, displays, motors together with wheels, arms, speaker

Original e-mail message from Gerhard Katstaller can be found as an attachment to this document.

1.4 Graphics User Interface

One Main window. The window is split into following frames:

- Drawing Screen

- Button bar

- Menu bar

- Drag and Drop

The Drawing screen is placed in the middle of the window and shows the to be designed agent.
The button bar holds several buttons:

- File items (new, load, save, merge(?))

- Control items (view(angle, zoom), properties, settings, ...

- Drag and Drop select (switch the drag and drop section between basic elements, sensors, actuators, ...)

The menu bar holds the program menu, from which it should be possible to use all commands of the program. Only the Drag and Drop cannot be implemented here.
The Drag and Drop section holds the basic structures, from which the robot will be designed. It is depending on the mode, which is set in the button bar. Available buttons depending on the mode are:

- Basic structures: sphere, cylinder, cube, ...
- Sensors: ir, light, bumper, sonic, ...
- Actiuators: light sources, wheels, ...
- ...

This is only the first draft of the GUI and is still to be developed!

1.5 Introduction to KIF

This was taken from: http://logic.stanford.edu/kif/Hypertext/node1.html

Knowledge Interchange Format (KIF) is a formal language for the interchange of knowledge among disparate computer programs (written by different programmers, at different times, in different languages, and so forth).
The definition of KIF is highly detailed. The following general features are essential in the definition of KIF.

1. The language has declarative semantics. It is possible to understand the meaning of expressions in the language without appeal to an interpreter for manipulating those expressions. In this way, KIF differs from other languages that are based on specific interpreters, such as Emycin and Prolog.

2. The language is logically comprehensive - it provides for the expression of arbitrary sentences in predicate calculus. In this way, it differs from relational database languages and Prolog-like languages.

3. The language provides for the representation of knowledge about the representation of knowledge. This allows us to make all knowledge representation decisions explicit and permits us to introduce new knowledge representation constructs without changing the language.

1.5.1 The KIF/ACL language

The following was taken from: http://dutcu15.tudelft.nl/ marcel/mapping/kif/top.html

The agent communication language (ACL) is centred around the knowledge interchange format (KIF). ACL/KIF was originally developed to exchange distinct information between applications,following the messaging paradigm, but due to the diversity of the tools to be integrated, translation was also a major topic of research. ACL/KIF identifies two kinds of translation problems:

- vocabulary translation as it arises from the differences of the abstractions inherent in the implementation of different agents (mapping in the sense of this paper) and
- the logical translation dealing with the consequences of the problems arising from the limits imposed by agents on the logical structure of messages.

The use of this language and the supporting agent-based architecture attempts to take software integration beyond the exchange of data files or the sole use of database management systems. As the language has to handle bidirectional mappings between the messages sent by the different agents in their native conceptualization, it canpartly be considered as a formal mapping notation (KIF).

Appendix A

Gerhard's letter

From: gerhard.katstaller@sbg.ac.at Thu Mar 11 12:34:15 1999 Date: Thu, 11 Mar 1999 07:53:30 +0100 From: Gerhard Katstaller ¡gerhard.katstaller@sbg.ac.at¿ To: Roland Schwaiger ¡rschwaig@cosy.sbg.ac.at¿ Subject: Re: PPKT II

[The following text is in the "iso-8859-1" character set] [Your display is set for the "US-ASCII" character set] [Some characters may be displayed incorrectly]

Anforderungen an das Roboter Construction Kit: (erste berlegungen dazu)
- Die Definition eines Agenten soll graphisch (3D) interaktiv erfolgen. Dies gilt auch fr physikalische Eigenschaften (z.B. die Winkelabhngige Empfindlichkeit eines Lichtsensors).
- bezglich Licht und Sound (Infrarot) sollte man sich ansehen wie das in Java 3D gelst wurde.
- Man sollte bezglich des externen Austausch-Formates sich auch die "Roboter Definition Language" ansehen (sofern es die irgendwo doch real (und nicht nur als Mythos:-) gibt).
- Das Konstruktion-Kit soll nicht nur fr Roboter (Agenten) verwendbar sein, sondern auch fr andere Elemente der Simulationswelt: Hindernisse, Licht, Ampel, Tonquelle, etc.
- Nachdem ein mit dem Konstruktion Kit erstellter Roboter spter damit auch wieder modifizierbar sein soll, ist es ntig, den aktuellen Zustand abzuspeichern und auch wieder einzulesen und daraus z.B. auch die 3D-Graphik wieder aufzubauen. Der abgespeicherte Zustand sollte sinnvollerweise auch dem Austauschvormat entsprechen. Das Einlesemodul soll auch im DiMAS einsetzbar sein.

- Wenn ein vernnftiges Format entsteht, dann kann ich eine Erweiterung davon als Speicherformat fr die DiMAS-Simulationen verwenden. (wegen Java 3D ist es sowieso sinnvoll ein anderes als das bisherige zu verwenden)
- Folgene Grundelemente frs Konstruieren fallen mir ein: Krper: Quader, Rechteck, Kreis, Zylinder, Pyramide, Kegel, Kugel, Torus, Rad, ...
- was ist mit bewegbaren Teilen wie Rder, Roboterarme, etc.?
- Attribute: Abmessungen, Position, Farbe, Transparenz, Reibung? (rauhe glatte Oberflche)

Sensoren: Licht, Infrarot, Wrme, Hearer
- Aktoren: Licht(Farbe), Ton, Wrme, Displays, Motoren in Zusammenhang mit Rdern, Armen., Speaker

Appendix B

Records

B.1 Briefings' Records

B.1.1 11.03.1999

Present are: SCHWAIGER Roland, KATSTALLER Gerhard, BRAUNEIS Wolfgang, ECKSCHLAGER Manfred, GANGL Thomas, HUSKIC Zenon, WOLACEWICZ Mateusz.
Time: 14 00 - 15 00.
Place: Briefing's room.
Introduction to the project and the first requirements;

As it was arranged on Wednesday, March 3, 1999, our team toghter with the coordinator met Gerhard Katstaller and Wolfgang Brauneis in order to get an insight regarding DiMAS. Members of the team got a copy of the e-mail message sent by Gerhard Katstaller to our coordinator with the first considerations listed. Gerhard and Wolfgang explained the purpose of DiMAS and set some other requirements for our project as well (movable parts of a robot, etc.). The Meeting resulted in the following items:

- Members of the team should get the DiMAS source code and "try" it out (source code should be sent per e-mail to each member from our coordinator)
- The next briefing should be held on March, 17, 1998 at 16 15 HRS (time has been fixed for Wednesdays at 16 15 HRS)

- Report regarding project should be submitted to the coordinator weekly, no later then 13 00 HRS Tuesdays
- Report should be written in a document preparation system: LaTeX

B.1.2 17.03.1999

Present are: SCHWAIGER Roland, GANGL Thomas, HUSKIC Zenon, WOLACEWICZ Mateusz.
Time: 16 15 - 17 00.
Place: Schwaiger's office.

- Discussing about the sturcture of the document
- Discussing content of the document
- Appending the e-mail message regarding the requirements
- Discussing new items

Modification of the document:

- It should get a title page
- Abstract must be imported
- Briefing's record should be prepared as appendix
- We should finally bring photos for the www-page

New items:

To introduction:

- What kinds of robots are there(mobile, static)?
- What kinds of sensors are there?

We should search in the Internet regarding RoCK (3D-game Decent should have something like that - Altavista) We should get in touch with Conrad Lang regarding the "Knowledge Interchange Format" (KIF). He made a work about it. Other sources are available at the Stanford University Server.
Design of the GUI should be started.
Requirements:

- It should be possible to use RoCK for other objects and not only for agents(robots).
- What attributes could be appointed to an object.
- Representing the system in the UML (Unified Modelling Language)
- Gerhard's e-mail message should be analysed once more

Bibliography

[1] P. J. McKerrow, *Introduction to Robotics.* Addison-Wesley Publishing Company, second ed., 1993.