Black Jack Dealer : A Robot that speaks and plays cards

Software Project SS 2004

Renjini Narendranath

Dept. of Computational Linguistics
Universität des Saarlandes
renjinin@coli.uni-sb.de

October 18, 2004

Abstract

In this term paper we present a robot that can play the card game of Black Jack against a user / player. We designed and developed within the duration of 7 weeks, a basic version of the Black Jack dealer to play with one player and then extended with one more module which can sort the cards at the delivery point. We also discuss some future extensions for the robot, so as to accomadate more card games.

1 Introduction

This project was conducted in association with the software project: Speaking Robots with Lego ® Mindstorms. Lego's Robotics Invention System (RIS) offers ample room for combining robotics with the idea of human-computer collaboration to achieve a particular goal. The aim of this project was to design and construct a robot that can engage with us in a natural language dialogue and collaborate with us for a task of certain complexity. The domain that we chose for this purpose was that of card-games. This was especially suitable to conceptualize and build a robot within the (hardware/software) limits of Lego building kit.

Though the basic kit provides a graphical programming system for the central processing unit (RCX), we used a flexible third party RCX programming called LeJOS [LeJ], [Bag02] which allows a fine-tuned programming using JAVA. LeJOS includes a Java Virtual Machine (JVM) that can interpret the byte-code which is compiled on the PC and then transported on to the RCX via an infra-red port.

Building 0 agents that can communicate with human beings in natural language is one among the goals of Computational Linguistics. In this project, we implemented such an agent that can engage in activities and collaborate with human beings. It is possible to extend the basic model of card-robot so that it can deal with more card games like Poker, Bridge etc. In this project, we tried to incorporate the image recognition software as well as a dialogue system that enables communication between the robot and the human player. The image recognition software was written by our team member Thobias Schild and the hardware part was constructed by Thomas Eigner.

Organization of this report

In section 2 we explain some of the basic functionalities of the Black Jack dealer. In section 3 we see the basic architecture of our Black Jack dealer. In section 4 we discuss about the dialogue based steering model of the dealer. In the subsequent section 5 we compare the performance of our card-robot with other robots like Luigi Legonelli and Bar-keeper Robot projects done by our colleagues as their respective projects. Finally, we conclude our results with section 6 and have a short discussion of the problems and their solutions that we encountered on the way.

2 Basic Functionalities of Black Jack dealer

In this section, we introduce the Black Jack dealer and discuss about its domain of activity. We then proceed to a rough outline of the game Black Jack.

Hallo, würden Sie ein Spielchen wa...gen ?
(Hello, would you play with me ?)

So begins the robot the deal of Black Jack game. The Black Jack dealer is unlike other robots, a static one. The domain of this robot for that matter does not require it to be mobile and go from one place to another. The human player that plays against this robot is the adversary in the game. The basic model of the robot is for the moment developed for playing the game of Black Jack, but this basic model can be extended in the future for playing more card games like Poker, Bridge etc. Figure 1 illustrates the state-of-art Black Jack dealer.

The Black Jack game

The game of Black Jack (also known as 21– game) is usually played in casinos where one plays against a dealer. Black Jack is played differently from place to place. We have adopted the following version of the game. Regarding the points, the cards with numbers from 2 to 10 take the respective numbers as the points. The pictured cards like hearts, spade and diamond holds 10 points each. If an ace is drawn in the beginning, that is considered to be a "soft-ace" and hence holds 11 points, and the ace's drawn later carries 1 point each. In the beginning, the player draws two cards and the points that he gets based on the drawn cards will be added. If the points sum up to 21, then the player has won the game in the first draw, if the sum is less than 21,

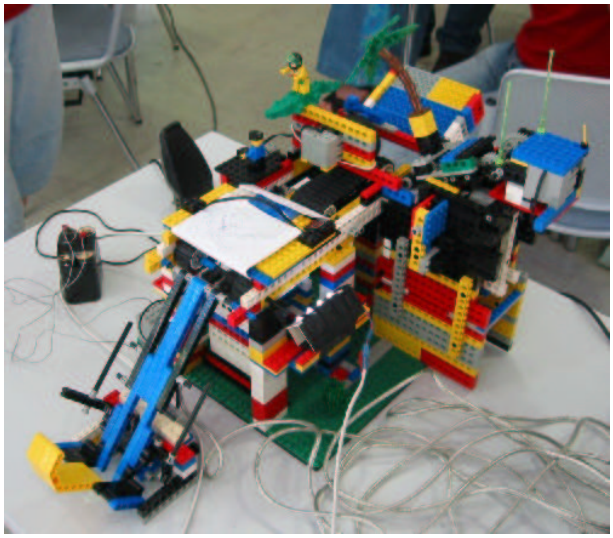

Figure 1:

then the player can take cards until he gets 21 or he decides to stand (stop drawing and waits for the dealer to make his move). If the player draws cards whose sum is more than 21, then he busts or lose the game. In the beginning, the dealer also takes two cards for himself. He reveals one card to the player, but covers the other one. When the player decides to stand, i.e, when he decides to stop drawing, then the dealer reveals his second card and draws cards one by one till he has 17 points. Then, he compares the points and declares who has won the game. If the dealer procures the highest points, then he wins and the player busts. In case, if both has equal number of points, then the game ends up in a tie, in such situation, the dealer wins automatically. An experienced player of Black Jack must know strategies like when to hit (draw) and when to stand making sure that the dealer busts.

Typically, in such games, betting of a fixed amount is involved. In our game, the player starts with a sum of 15 Euro which he can bet as a whole, or small dividends of money throughout the game. Of course, the idea of betting the money is fictious and for the moment, we don't really hand over the money to the dealer.

3 The Architecture of Black Jack dealer

The robot has four different modules in addition to the dialogue module. Most of the modules except the camera are operated by motors. The movements of the motors as well as the camera are in turn controlled by the RCX

box.

The Card Stack / spender module : This module holds a stack of cards and with the help of a motor, cards are issued one by one to the conveyer belt which conveys those cards to the player. Initially, this module is loaded with shuffled cards. The number of cards to be issued each time is controlled by the RCX programme.

The conveyer belt with embedded camera : The card issued from the card stack / spender module is transported to the other end of the machine where the player is stands, with the help of two conveyer belts. We designed two belts instead of one, because of the hard ware limitations. A camera is embedded in one of the conveyer belts. This camera focuses on the cards that travels through the belt and reads the bar-codes on the card.

In figure 2 we present the overall architecture of the card-robot. We have left out the four different motors in the picture. *The RCX and motors* : The RCX

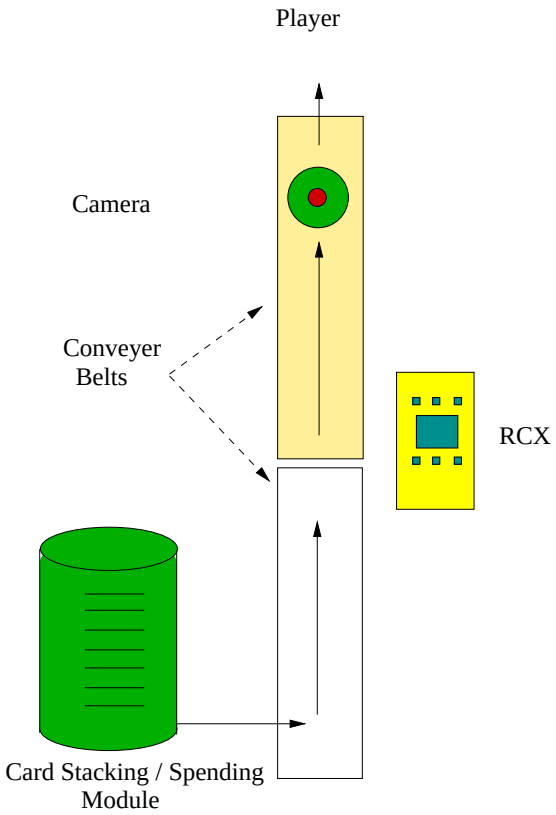

Figure 2: The architecture of Black Jack Dealer

platform controls the different motors. The RCX contains programmes which

specifies which among the three motors should move forward / backwards, or stop moving altogether. The movements of the motors result in the rotation of the conveyer belts or in the issuing of cards by the spender module.

The Opener Module: This module was done as an extension of the basic architecture as shown in figure 2. This module is responsible for flipping open the cards transported through the conveyer belts. This module takes into account that, the second card of the dealer should not be i revealed to the player initially. All the other cards are flipped open soon after they slide in to this module.

In addition to the hardware, we also designed the playing cards with bar-codes and got them printed on paper cardboards. A sample card is shown in the figure 3 where we can see the upper portion with the logo "Robo Cards" and the lower portion of the card with normal number and sign along with the bar-code.

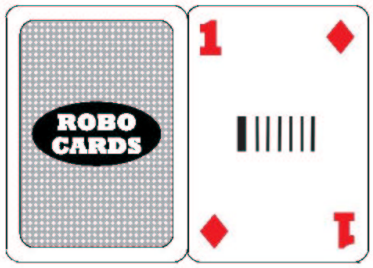

Figure 3: A sample card

4 The Dialogue based steering of the Robot

The dialogue plays a vital role in the game with the dealer. The speech input is recognized by L & H recognizer client to which we specify a predetermined set of grammar that contains the exact sentences that we are want to speak to the dealer. This grammar is written context free. For modeling the dialogue for the Black Jack dealer, we used the commercial software called Diamant Dialogue System developed at the CLT Sprachtechnologie GmbH. The game strategy used for the Black Jack dealer is encoded in the Diamant dialogue manager module. We specifed dialogues using the GUI interface provided by the Diamant with the help of nodes and edges connecting them. Each node represent a state in the dialogue. The dialogue thus specified is stored in xml format files. The Diamant dialogue module allows socket based connection to multiple clients. In our case, we had plugged in the camera client RUDI which recognizes the bar-code image on the card and communicates with the

Diamant the number points. The structure of the dialogue system is outlined in figure 4.

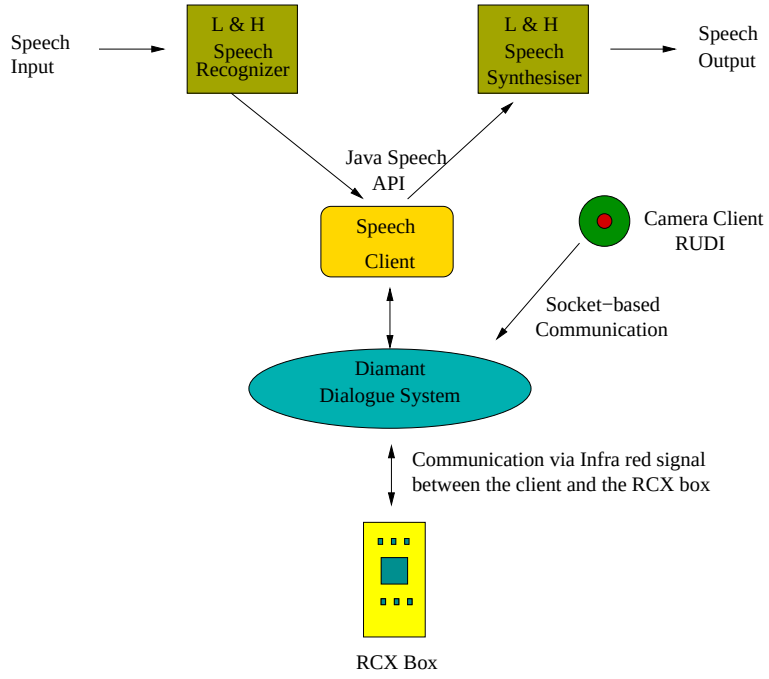

Figure 4: The dialogue system

The dialogue designed for our Black Jack dealer proceeds in a deterministic finite state fashion. That is, we can specify the dialogue moves according to our needs and therefore make the system behave deterministically. The main dialogue moves in the game are based on the rules of the game as outlined in the section 2. After the system is initiated, the dialogue opens with a greeting and asks if the user wants to play with the dealer. On getting a positive reply, the dialogue proceeds to the game where it orders the motors in the card stacking / spending module, to issue the initial cards for the player and for the dealer himself. Figure 5 shows the main dialogue graphs with the different nodes including a node named Graph Whole Rules. This is a sub-graph that contains the whole array of nodes capturing the different rules and moves of the game. A snapshot of the sub-graph is shown in figure 6

The dealer issues further cards upon request from the user. Each card passing past the camera embedded in the conveyer belt is recognized and the points are memorized in variables in the image recognition software written by Thobias. At the end of game the points are added and communicated to the user again through the dialogue manager.

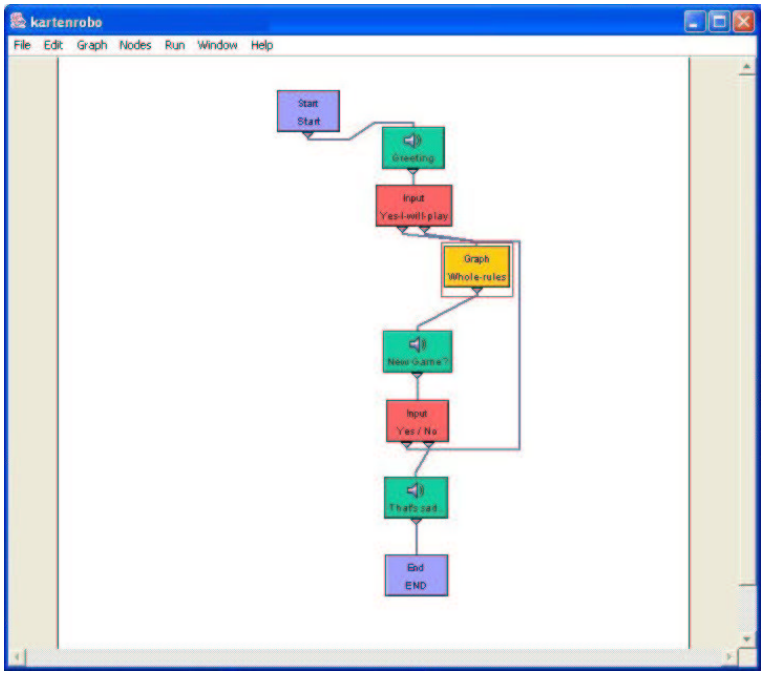

Figure 5: The main dialogue graph containing the minimal number of nodes

5 Comparison of Black Jack dealer with some other Lego-Speaking robots

In this section, we compare our Black Jack dealer with other robots of the co-projects in this course. [KK04] provides an account of different robotics projects that were taken up by students in the year 2002.

The Black Jack dealer resembles the Chess-Player robot in the domain of toy games. The dialogue model used by the Chess-Player robot has added feature of cross checking and verification of the speech input. In our dealer, we encountered problems of verification during the image recognition phase of the bar-codes in the cards. This was eventually solved through the dialogue by urging the user to feed-in the card once more. The Chess player robot employs the game strategy based on the GNU-chess game client.

In contrast with the chess-Player robot, the Black Jack dealer offers a very short game which can be played in few minutes. This aspect of the Black Jack game, renders possibility for writing short speech recognition grammars and in modelling short dialogue.

The pyramid explorer robot Indiana, also used a camera like the Black Jack dealer. The camera there, was not used for any kind of image recognition unlike our robot, as it only transmitted images to the user. Though Indiana

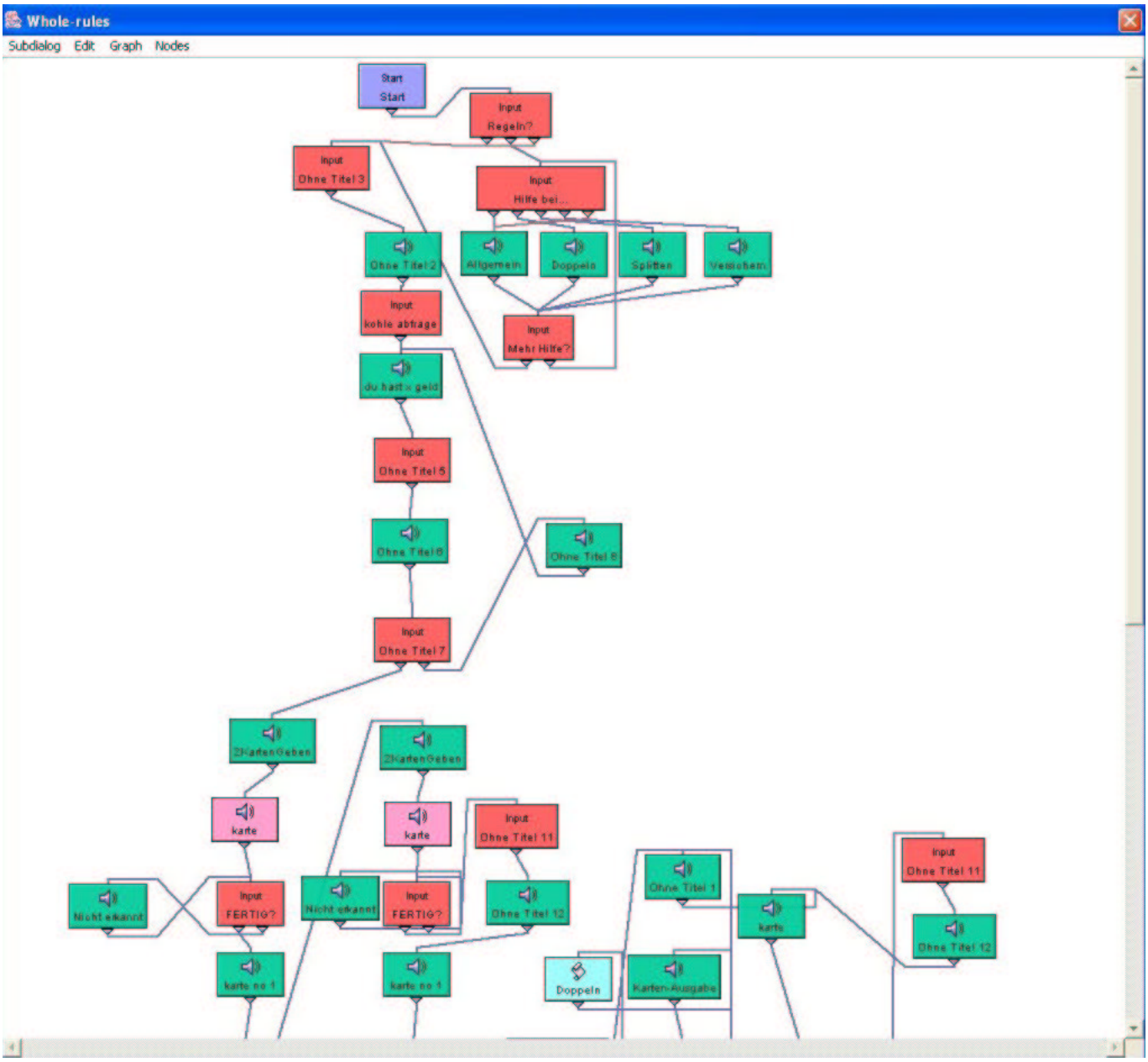

Figure 6: Part of the subgraph where the rules of the game are captured in the nodes

did not have to face problems with image recognition, they had encountered additional problems of orienting the robot because it was mobile and moving, and had to find a way through convoluted passages.

Luigi Legonelli and the Barkeeper robots were both robuster with their dialogue systems. The Barkeeper robot was equipped with a MySQL database which provided the user to define his favorite drink and save the recipe for future use.

6 Conclusion

In this project we designed and implemented a card playing robot: the Black Jack dealer. We could successfully incorporate the dialogue based communication between the user and the robot. Though we encountered at times problems with the image recognition software of the camera, which was mainly due to the amount of lighting needed from place to place, we solved it by installing LED bulbs which provided sufficient lighting.

We extended the hardware such that, in case if the bar-code in the card is not recognized, then the camera client RUDI would send an error message to the dialogue manager. The dialogue manager then asks the user to feed the card from an independent opening in the conveyer belt so that the camera can read the card once more.

The dialogue modeling for this robot has taken into account the possibilities of being informative and collaborative. Firstly, the user is given an option of getting informed about the rules of the game. Nevertheless, this option is only available towards the beginning of the game. One reason for doing this was to avoid the potential blow-up of the nodes in the dialogue graph. We should also point out here that, as the speech recognizer could accept only the inputs in German language, a non-native speaker could face problems with the speech recognition software. A typical example would be when the user / player wants to bet *five* Euro, which in German is *fünf* Euro. Typically if the non-native speaker cannot pronounce the *ö* diphthong of German, in many cases, the recognizer recognizes a completely different number.

The dialogue model used in our project is elegant and simple, but has potentials for including wits and jokes like in Luigi Legonelli robot. Another possible extension for future is by extending the robot with a hand module, where the dealer can stack his own hand of cards. Thomas had tried to build a hand module which could hold 5 cards, but even then, due to the innate hardware complexity, the module was quite large in size. Therefore, we had to put off the very idea due to sparseness of Lego bricks for building a hand

module that could hold 30 or 40 cards.

References

[Bag02] Brian Bagnall. *Core Lego Mindstorms Programming*. Prentice Hall PTR, 2002.

[KK04] Alexander Koller and Geert-Jan Kruijff. Talking robots with lego mindstorms. In *Proceedings of the 20th COLING*, Geneva, 2004.

[LeJ] LeJOS. Java for rcx http://lejos.sourceforge.net/.