# Statistics-Based Cognitive Human-Robot Interfaces for Board Games – Let's Play!

Frank Wallhoff, Alexander Bannat, Jürgen Gast, Tobias Rehrl, Moritz Dausinger, and Gerhard Rigoll

Department of Electrical Engineering and Information Technology,
Institute for Human-Machine Communication
Technische Universität München
80290 Munich, Germany

**Abstract.** The archetype of many novel research activities is called cognition. Although separate definitions exist to define a technical cognitive system, it is typically characterized by the (mental) process of knowing, including aspects such as awareness, perception, reasoning, and judgment. This especially includes the question of how to deal with previously unknown events.

In order to further improve today's human-machine interfaces, which often suffer from deficient flexibilities, we present a cognitive human-robot interface using speech and vision. The advancements against regular rule-based approaches will become obvious by its new interaction strategies that will be explained in the use case of a board-game and a robot manipulator.

The motivation behind the use of cognition for human-machine interfaces is to learn from and adapt to the user leading to an increased level of comfort. For our approach, it showed proof that it is effective to separate the entire process into three steps: the perception of external events, the cognition including understanding and the execution of an appropriate action.[1]

## 1 Introduction

The excellence research cluster *Cognition for Technical Systems CoTeSys* is working on establishing a more intelligent and useful behavior of technical systems, in particular for unpredictable events in its surroundings. Therefore, cognition [1,2] could be the way to surpass the current constraints (rule-based and static behavior) incorporated in nearly all kinds of today's operating technical systems.

Such a cognitive system will monitor its environment and respond on changes. Furthermore, its skills will improve over time using generic learning algorithms.

In general there are many possibilities where technical systems equipped with cognition can cover new areas of applications. The cluster of excellence focuses on ambient living [3] and advanced robotics [4], especially in an industrial context [5].

[1] All authors contributed equally to the work presented in this paper.

M.J. Smith and G. Salvendy (Eds.): Human Interface, Part II, HCII 2009, LNCS 5618, pp. 708–715, 2009.

However, the before mentioned two scenarios (ambient living and advanced robotics) entail many restrictions and constraints due to the technical state of the art as well as the field of application, therefore, we consider to work on a further field of application focusing on pure cognitive processing (playing games).

The rest of this paper is organized as follows: In Section 2, a brief description of the hardware set-up in action is given. In Section 3, we introduce the underlying system architecture for the presented cognitive game application. Afterwards, in Section 4, we have a closer look on the actual interaction between the humanoid player and the cognitive playing counterpart with its actuator in form of an industrial robot manipulator arm. The paper closes with a summary and an outlook over the next planned steps.

## 2 Hardware Set-Up

The constructed hardware set-up is built by two major components. First, the actuator used to interact in the board game is an industrial robot arm. Second, for the presentation and scene observation a framework is mounted above the game interaction plane. This plane is equipped with the industrial robot arm. Besides, the necessitated computations for the cognitive game playing interactions and manipulation of the pawns are executed in a distributed manner, involving two processing units (standard PC with AMD Phenom 2.2GHz quad-core and four gigabyte RAM). One of these units is exploited to allow remote access towards the robot controller as well as a displaying module. The remaining unit is in charge of the cognitive game processing as well as scene surveillance.

### 2.1 Cognitive Interaction Unit

The industrial robot manipulator arm currently involved in the cognitive game playing process, is a Mitsubishi robot RV-3SB. This robotic interaction unit has one arm with six degrees of freedom and it can lift objects with a maximum weight up to three kilograms. The effective radius for performing gripping interactions is limited to a total of 0.642 m around the robot's trunk. Its tool point is equipped with a force-torque-sensor and a tool-change unit.

The instantaneous communication between the robotic arm manipulator and the game control unit is implemented via a network interface provided by a middle-ware similar to CORBA. This approach facilitates the interaction between different operating systems as well as different programming languages. Additionally, the outsourcing of the robotic control unit in an own module allows for an easy substitution and integration of other robotic manipulators in the future. A client-server model has been chosen as communication paradigm. The implemented server provides the client with three dimensional access to all reachable points of the robot arm. However, for the game operation we constrained the allowed movements in two parallel x-y-planes, see Figure 1. One plane is used for the translation of the pawns from one game field point to another. The pick-up and place operation is performed in the second movement layer, closer to the actual game plane.

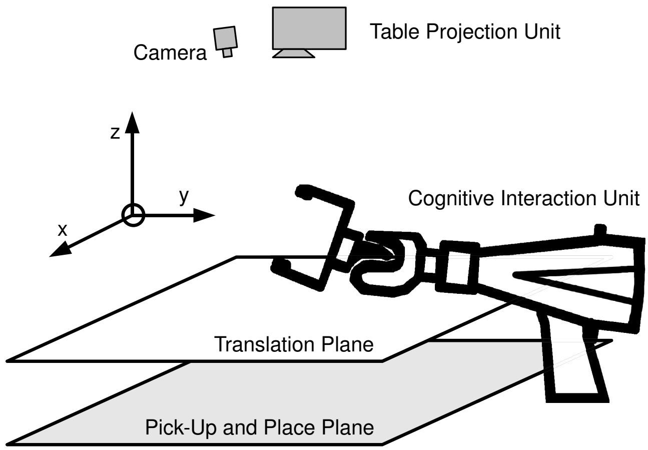


**Fig. 1.** Overview of the working x-y-planes

### 2.2 Projection and Observation Units

A projection unit is attached in the framework mounted above the interaction plane. Via this projection unit the board game field and required interaction buttons are displayed onto the workbench at a total resolution of $1024 \times 768$ pixels. The obeyed vision sensor consists of a regular webcam with a resolution of $640 \times 480$ pixels running at 15 frames per second. Furthermore, a microphone with sufficient sound quality for speech recognition is also embedded in this device leading into a handy setup. These sensors allow for the recognition of speech commands, the recognition of movements and actions on the desk as well as the detection of gaming pieces and the number of eyes on the dice.

## 3 System Architecture

The connection and interaction of the above described sensors and actuator is formed by the well structured system architecture. This architecture has a modular concept, basing on the approach presented in [6,7,8]. The therein presented real-time database is capable of handling huge amounts of data from different input modules, having a different update rate. The desired module interaction is achieved together with the Internet Communication Engine (ICE) middle-ware. We will have a closer look on these participating modules in the following.

### 3.1 Dialog Manager

The management and adjustment of information exchange between the user and the system is realized by the dialog manager. In this case we use a commercial version of a dialog management system for speech recognition and speech synthesis. According to the current step within the game and perceived events, the system can utter information for the user or comment on these events. The speech recognition is primarily required at the start of the game for initializing the interaction with the player. Therefore, simple decisions have to be made, e.g. which color the pawns of the player should have. In these first interaction steps, the necessitated grammar for the speech recognition can be adapted in run-time to enhance the recognition rate. Afterwards, the state machine will transit into the playing mode, where the game logic and the related robotic actions are performed. Therefore, to enable a cognitive playing interaction, the "Cognitive Playing Strategy Module" takes over the control to create reasonable playing moves.

### 3.2 Cognitive Playing Strategy Module

This module is responsible for determining the next playing moves. At first, the rules of the board game "Mensch ärgere dich nicht" have to be incorporated into the knowledge database. This database is composed of a PROLOG-based rules collection, which can be accessed via a C++ wrapper. The database is managed and kept up-to-date with current positions of the pawns to infer the next possible allowed moves. After extracting all allowed possible moves, a Bayesian Network is used to determine the best choice. Since the training of the Bayesian Network is entirely done by data retrieved from previous games, the inference will improve its performance continuously. With this gained result the related gripping and movement actions are conducted by the robot.

### 3.3 Displaying Module

The content shown by the projection unit is controlled by the displaying module. This module is implemented as a server and offers remotely callable procedures to display images at given coordinates and size onto the workbench. Highlighting of objects and special regions for interaction, e.g. for rolling the dice, can also be done with the incorporated functionalities. Finally, this module can also project Soft-Buttons from above for interaction purposes. The content of these buttons can be freely chosen and their location can be arbitrarily selected. The spatial information about these buttons is written back into the real-time database for the image processing based detection of button activation.

### 3.4 Dice Recognition Unit

For gaining more natural feelings and moods in playing with the here presented system, a dice recognition unit is also implemented. The dice has to be tossed in a dedicated area on the game surface, highlighted with the table projection unit. In the following, the required steps are described more precisely.

1. First of all, the user has to toss the dice.
2. In the first processing step, the top plane of the dice is recognized. This is performed by applying a thresholding filter operation on the observer image. We are currently using a dark dice with light eyes on a white surface. Therefore, the dark area of the dice can easily be extracted from the filter

image. After applying morphological operations to the filter image, only those areas remain in the filter image, that contain the dice.

3. In the next processing step, the before extracted dice area is further analyzed. The inverse filter operation of the above described is applied to the dice region. Additionally, noisy elements in the filter image are removed. At this step, only those surfaces are visible, which are related to the number, that has been tossed.
4. At this stage, the filter image can further be analyzed to detect the actual number of eyes. Due to the chosen setting, only the number of eyes on the dice are visible in the filter image. By applying the connected components algorithm [9], these areas are labeled. By counting the amount of labeled surfaces, one gets the actual tossed value of the dice.

### 3.5 Robot Server

This module is working on the second computing unit and offers an interface to control the robot. However, it is not only a simple server performing all received orders, but it also provides for a certain degree of secure handling, because it prevents harmful commands resulting in damage of the hardware set-up. In addition, the system can be interrupted at any time by the user and will continue with the last requested action on demand.

### 3.6 Controlling Unit

Since all participating modules are built as stand-alone units, a central controlling unit is implemented to keep track of all relevant game data. The main task of this unit is to synchronize data streams provided by the real-time database with delays caused by the user and the movement of the robot. Besides the current game status, adjustments to the visual output like scaling or rotation are stored in this module as well. Thereby, the robot server can proceed the actual movement based on the transformation matrix which is used for the visual output.

## 4 Let's Play

In the first of the subsequent two sections, we will present a short overview on the game constituting the foundation for the cognitive interaction scenario, where a cognitive computer system takes part in a contest in playing "Mensch ärgere dich nicht" against a human player, see Figure 2. In the Section 4.2, the required adaptations for the realization of the presented human-machine contest will be delineated in greater details.

### 4.1 Rules

"Mensch ärgere dich nicht" is a classical German board game. It exhibits similarities to the Indian game *Pachis*i, the American game *Parcheesi*, and the English game *Ludo*.

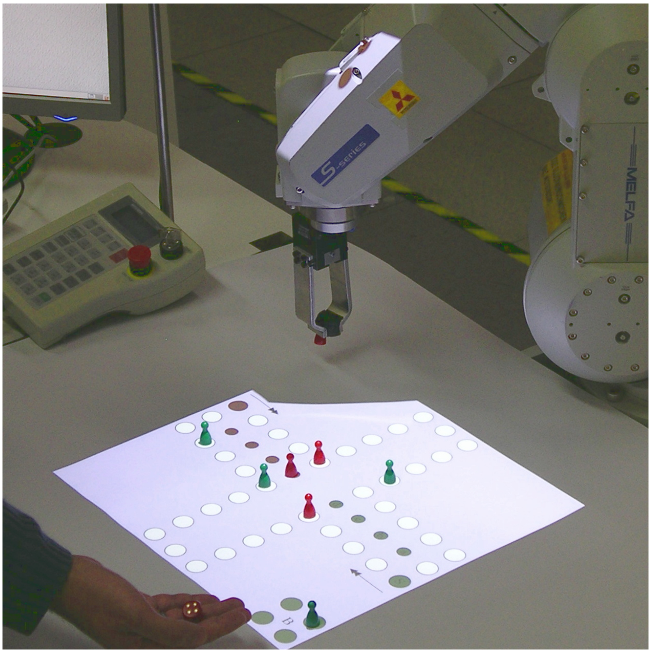


**Fig. 2.** “Mensch ärgere dich nicht” with a robot

In the classical style a wooden board is used for the game interaction. On this board all pawns are placed. For each player there are four pawns, which are color-coded (e.g. red, green, blue, yellow) to avoid any possibilities of confusion in the playing.

The game can be played by 2, 3 or 4 players – one player per board side. At each board side a so-called home area is situated. This home area is the starting place, where the four pawns of each player are placed at the beginning of the game.

The objective of the game is the following: Every player tries to bring the four pawns into the so-called “home row”. The starting point for each player is the home area. The player has to pass his pawns clockwise around the game board from the home area into the home row.

One player starts to toss a dice. If he tosses a six, he will put one of his four pawns on the “start field”. The player has to toss again, and the number of eyes of the dice determine the fields, which he can pass with his pawn. If another pawn of an opponent player is at the destination field, this opponent pawn has to be moved back to its home area again.

If he tosses no six, he can toss the dice twice again to obtain a six. If no six has been tossed, the turn moves over to the next player.

### 4.2 The Robot Way of Playing

Due to the today's state of the art of robotic gripping and observation techniques, some constraints and restrictions are imposed on the current interaction setup. First, the robot is unable to toss a dice, therefore, a random number generator is delivering the robot's numbers for the moves in the board game. Second, because of the small size of the pawns and the current vision-based surveillance all moves in the board game (robot's and player's) are conducted by the robot. Therefore, a touch screen display is used as input for the human's moves. The player selects a field of the board game displayed at the touch screen and thus initiates the according movement of the pawn carried out by the robot. Besides, the board game is projected via the table projection unit.

## 5 Conclusion and Future Work

In this paper we presented a new form of a cognitive interaction scenario. The focus was laid on the cognition required for a human-machine contest in playing the German game "Mensch ärgere dich nicht". The implementation of the cognition playing strategy exhibits great potential for becoming a dignified counterpart for a human. However, in the current set-up, there are still many constraints, we would like to surpass in the future. First, the human should be allowed to move the pawns on his own, thus enabling him to cheat as well as to bring the system disadvantages by misplacing the pawns of the robot. For the realization of this vision, it is absolutely mandatory that the vision-based pawn detection and recognition has to be improved tremendously. On the other side, the cognitive system should attempt to betray the player as well from time to time for improving his winning opportunities and making the game more natural.

## Acknowledgment

This ongoing work is supported by the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see www.cotesys.org for further information and details. The authors further acknowledge the great support of Matthias Göbl for his explanations and granting access to the RTDB repository. Furthermore, we want to thank all our partners within our CoTeSys-Projects for the fruitful discussions and implementation work to make our visions and ideas become reality.

## References

1. Vernon, D., Metta, G., Sandini, G.: A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. IEEE Transactions on Evolutionary Computation 11(2), 151–180 (2007), http://dx.doi.org/10.1109/TEVC.2006.890274

2. Anderson, D.M.L.: Embodied cognition: A field guide. Artificial Intelligence 149(1), 91–130 (2003), http://cogprints.org/3949/
3. Beetz, M., Stulp, F., Radig, B., Bandouch, J., Blodow, N., Dolha, M., Fedrizzi, A., Jain, D., Klank, U., Kresse, I., Maldonado, A., Marton, Z., Mösenlechner, L., Ruiz, F., Rusu, R.B., Tenorth, M.: The assistive kitchen — a demonstration scenario for cognitive technical systems. In: IEEE 17th International Symposium on Robot and Human Interactive Communication (RO-MAN), Muenchen, Germany (2008) (invited paper)
4. Lenz, C., Suraj, N., Rickert, M., Knoll, A., Rösel, W., Bannat, A., Gast, J., Wallhoff, F.: Joint actions for humans and industrial robots: A hybrid assembly concept. In: Proc. 17th IEEE International Symposium on Robot and Human Interactive Communication (August 2008)
5. Zäh, M.F., Lau, C., Wiesbeck, M., Ostgathe, M., Vogl, W.: Towards the Cognitive Factory. In: Proceedings of the 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV), Toronto, Canada (July 2007)
6. Goebl, M., Färber, G.: A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles. In: Intelligent Vehicles Symposium, pp. 737–740 (June 2007)
7. Stiller, C., Färber, G., Kammel, S.: Cooperative cognitive automobiles. In: 2007 IEEE Intelligent Vehicles Symposium, June 2007, pp. 215–220 (2007)
8. Thuy, M., Göbl, M., Rattei, F., Althoff, M., Obermeier, F., Hawe, S., Nagel, R., Kraus, S., Wang, C., Hecker, F., Russ, M., Schweitzer, M., Leon, F.P., Diepold, K., Eberspächer, J., Heißing, B., Wünsche, H.-J.: Kognitive automobile - neue konzepte und ideen des sonderforschungsbereiches/tr-28. In: Aktive Sicherheit durch Fahrerassistenz, Garching bei München, April 7-8 (2008)
9. Jankowski, M., Kuska, J.-P.: Connected components labeling - algorithms in mathematica, java, c++ and c-sharp. In: Mitic, P., Jacob, C., Crane, J. (eds.) New Ideas in Symbolic Computation: Proceedings of the 6th Mathematica Symposium, Positive Corporation Ltd. Hampshire, UK (2004)