

cljRobust - Clojure Programming API for Lego Mindstorms NXT

Konrad Kułakowski

Institute of Automatics,
AGH University of Science and Technology
Al. Mickiewicza 30,
30-059 Cracow, Poland
`konrad.kulakowski@agh.edu.pl`

Abstract. This paper presents the *cljRobust* - middleware library, which allows an intelligent control application to be written for *Lego Mindstorms NXT*. *cljRobust* is based on *Clojure*, which is a modern, concurrent dialect of the *Lisp* language. It shortens the distance between deliberative, symbolic processing and hardware. In this way, *cljRobust* facilitates the writing of control algorithms which operate close to the hardware layer, and thus are able to quickly respond to changes in the environment. Since *Lego* robots have been adopted by many leading academic centers all over the world as a standard tool for teaching robotics, the author hopes the *cljRobust* library may also be found as an interesting teaching aid in AI robotics.

1 Introduction

One of the important fields of research in AI is the building of intelligent, autonomous robots. It requires knowledge of many different disciplines, such as control theory, mechanics, computer science, artificial intelligence and others depending on the robots' purposes. Formerly, such kind of constructions were available only for a limited number of scientists in several leading research centers. Over time, intelligent robots have become more and more popular. These days they are available for non-experts in the form of out-of-the-box, ready-to-use robotic sets provided by independent manufacturers. A good example of such robotics solutions is *Lego Mindstorms NXT* - a didactic tool recognized by many colleges and universities all over the world [5,27]. Although *Mindstorms NXT* has a lot of fans, they have to deal with several major drawbacks of the NXT platform, where the two most important are: limited hardware resources of the platform and inefficient bluetooth based communication NXT-PC. In response to these problems the *Robust* library has been proposed [21]. This addresses the issues by moving CPU intensive processing to a PC platform and providing a robust and efficient communication.

This article presents the *cljRobust* - the proposition of *Clojure* contribution library, which facilitates the programming of *Lego Mindstorms NXT* with *Clojure*

- a new, efficient *Lisp* dialect running on a *Java Virtual Machine* [16]. The combination of Lisp's expressiveness with Java's concurrency and its large standard library provides a powerful tool to the Mindstorms' programmers, which allows them to implement complex, intelligent, CPU intensive control algorithms. Lisp's roots of *Clojure* facilitates a symbolic knowledge processing. *cljRobust* is a part of a larger project conducted by the author aimed at providing portable and flexible intelligent control system architecture for autonomous mobile robots [22].

The introductory section of the paper contains a short outline of AI Robotics including remarks concerning *Lisp* usage in AI robot programming. There is also a short note about the *Clojure* language. In Section 2 the existing programming approaches for *NXT* are briefly summarized, including usage of the *Robust* library. Section 3 contains the proposal of a *Clojure*-based API for *Mindstorms NXT*. Section 4 presents a simple control algorithm written in *Clojure*, and, finally, section 5 includes a summary and the plan for future research.

1.1 AI Robotics

As an example of one of the early works in the field of AI Robotics, we may consider HILARE - a mobile robot constructed in LAAS (*Laboratoire d'Architecture et d'Analyse des Systèmes*) [15] and *SHAKY* the robot constructed by Nilsson [26] at *Stanford Research Institute*. Over the years these pioneer intelligent control systems evolved into three major paradigms of AI robotics [3,25]:

- Reactive/Behaviour Based Approach
- Hierarchical Planning and Approach
- Hybrid Approach

One of the most well-known advocates of the first approach is R. Brooks [7]. According to the principle "The world is its own best model" [8] reactive/behaviour based control systems focus on data sensing and behaviour generation rather than elaborative and time consuming knowledge analysis. This kind of systems is useful for rapidly changing and unstructured environments, where there is no time for complex knowledge processing.

Another approach was adopted by J. S. Albus and A.M. Meystel [2]. In the work "*A Reference Model Architecture of Intelligent Control*" they proposed a hierarchical control system intensively using the knowledge base and world model. In their system, every component has its own computational nodes, which are organized into hierarchical layers. Every layer has its own, well defined area of responsibility and specific timing.

An attempt to reconcile of the two previous approaches is the hybrid paradigm. It tries to take benefits from both hierarchical planning and the reactive control paradigm. As a result, the hybrid system sometimes behaves in a reactive manner, and at other times performs careful knowledge-based hierarchical planning. *AuRA* [4] or *PRS* [14] may serve as examples of this class of solution.

There are many languages that are used for programming robots. One of them is *Lisp*. Probably the first reference showing usage of Lisp-based language