

Context and Learning based Approach to Programming of Intelligent Equipment

Andrey V. Gavrilov

*Department of Automation of Production in Machine Building,
Novosibirsk State Technical University
Karl Marx Av., 20, Novosibirsk 92, 630092 Russia
Andr_gavrilov@yahoo.com*

Abstract

In this paper novel approach for programming of intelligent equipment based on usage of context and learning is proposed. Main feature of this approach is opportunity of combination of programming and learning, fuzzy and deterministic representation of knowledge and routine in one system. The elements of Context Based Language for Robot (CBLR) applying this approach are described.

1. Introduction

One of most actual problems in development of intelligent systems for manufacturing is human-machine interface. Two kinds of such interfaces are known, oriented on programming and learning respectively. Programming is used usually for industrial robots and other technological equipment. Learning is more oriented for service and toy robotics.

There are many different programming languages for different kinds of intelligent equipment, for industrial robots-manipulators, mobile robots, technological equipment [1, 2]. The re-programming of industrial robotic systems is still a difficult, costly, and time consuming operation. In order to increase flexibility, a common approach is to consider the work-cell programming at a high level of abstraction, which enables a description of the sequence of actions at a task-level. A task-level programming environment provides mechanisms to automatically convert high-level task specification into low level code. Task-level programming languages may be procedure oriented [3] and declarative oriented [4, 5, 6, 7] and now we have a tendency to focus on second kind of languages. But in current time practically all programming languages for manufacturing are deterministic and not oriented on usage of learning and fuzzy concepts like in service or military robotics.

But it is possible to expect in recent future reduce of difference between industrial and service applications. For example, known plans of Toyota Inc. to employ humanoid robots in automobile manufacturing. So it seems interesting and perspective to apply achievements of AI and Computational Intelligence in development of programming languages for industrial applications such as industrial robots or technological equipment.

In this paper we suggest novel approach to programming of robots and technological equipment based on usage of context and learning and so oriented on usage of natural language for learning. In this approach we do not distinguish learning and programming combining in one system declarative (description of context) and procedural knowledge (routines for processing of context). Also here new Context Based Language for Robot (CBLR) is proposed and elements of one are described.

This suggested approach is based on research of author during development of system software for transportation industrial robots based on learning [8, 9] and development of software for searching of documents by natural language query [10].

2. Previous example of usage of similar approach for programming of transportation industrial robots

In end of 80th years some of main principles proposed in this paper were used by author for development of dialog system software for programming of transportation robots in flexible automated production of capacitors [8, 9] (in Russian). This transportation robot-manipulator aims to move special cassette within half-finished capacitors between cells for different technological operations.

A basic language for robot programming (BLRP) was developed based on usage of context. General feature of this language was that there was only one simple command for action without any parameters. All parameters for action were obtained from context determined by another commands. And these parameters may be defined in arbitrary order. Thus this language was oriented to utilize natural language for learning and programming of robot.

The language BLRP consisted of following groups of commands:

- 1) For assignment of context,
- 2) For control of moving,
- 3) For control of execution of routines,
- 4) For definition of conditions,
- 5) For determination of points in environment (service area) of robot,
- 6) For description of state of robot and technological equipment,
- 7) Serving commands.

Architecture of software developed for robot consists of subsystems for programming, checking and execution of routines (figure 1).

Context included main following parameters:

- 1) X, Y, Z – coordinates of point,
- 2) Description of condition operator,
- 3) State of equipment,
- 4) State of robot,
- 5) Number of link of robot,
- 6) Number of cell of equipment,
- 7) Number of object.

Number in these parameters means identifier.

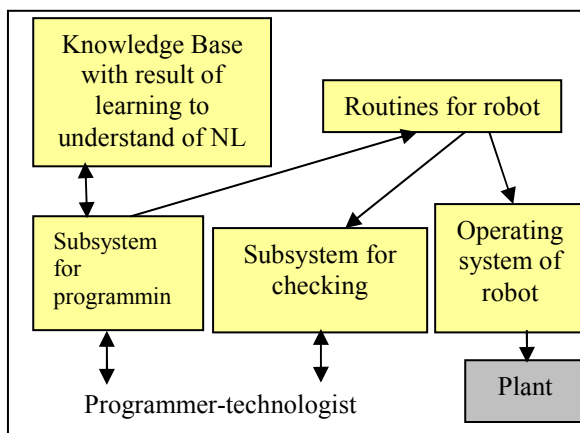


Fig. 1. Architecture of software for transport robot.

Programming in this system was replaced by learning. Result of one was knowledge base with associations between words of natural language and sequences of commands for description of context and

command “act” for start of action (move, pick or break with respect to context).

Unfortunately this system was not completed and tested in manufacturing because objective reasons in economy of Russia about 1990th year. We had only simulation of this system. But skill with this development and experiment shows that this approach has great potential. So now it is reasonable to return to this approach with new ideas and platforms.

3. Context and learning based programming language

We suppose that context and learning based programming language must be satisfied to following requirements:

- 1) This language is interpreted.
- 2) Modules based language.
- 3) It is oriented on parallel or simulated parallel execution of modules.
- 4) It is oriented on fuzzy and propositional logic.
- 5) Context is a set of different variables generated by different modules and needed for execution of other modules.
- 6) Context may be not full for any module and this module can be executed with respect to uncertainty of context.

7) This language is opened for creating and improving of semantics by including of interpretation of operators by statements in lower programming language for corresponding equipment. In this case our language may be viewed as macro processor or preprocessor for that language.

These features of our language allow using it for robots at dialog with in natural language for learning and programming of one, for programming of behavior of smart objects in smart environment [11] and so on. Our paradigm of this approach is illustrated by figures 2 and 3.

The programming in our proposed language is a process of definition and assignment of context variables during learning by presentation of associations between them and natural language words. This learning is similar to one implemented by author in software for searching of documents [10]. At the same time any context variable may be determined by procedure implemented in lower level programming language using in control system of robot or in another embedded control system. Let name this lower language as Context Based Language for Robot (CBLR).

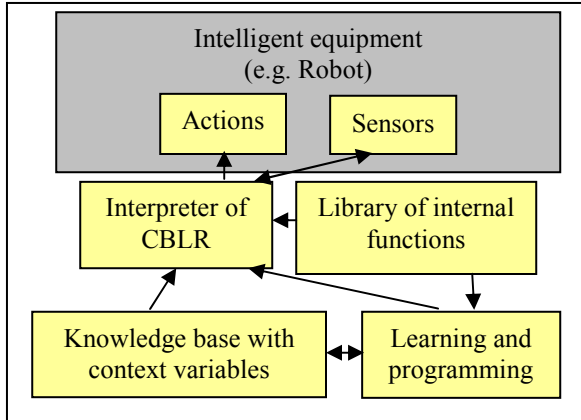


Fig. 2. General paradigm of learning/programming of intelligent equipment. Components.

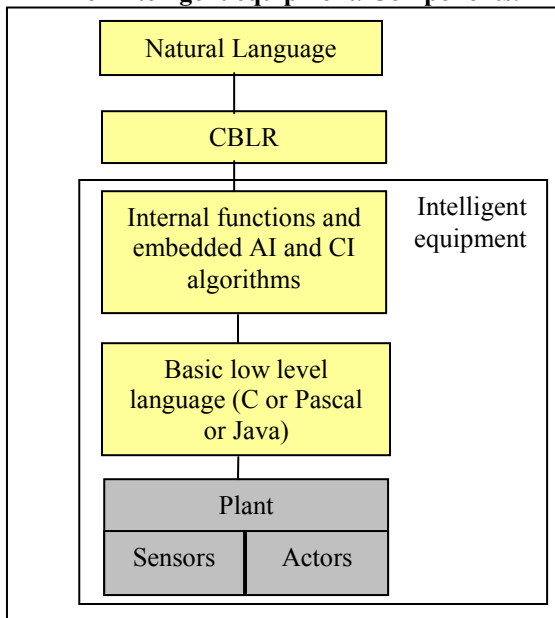


Fig. 3. General paradigm of learning/programming of intelligent equipment. Levels of languages.

An appearance of variable in routine of CBLR means that this variable gets any memory and became available for commands. For example, “#near_door” means that we began to work with variable “near_door” and value “true” is assigned to it. But besides this variable may be used as variable with numerical value determined later in routine (like “#near_door := 30”). Last statement may be used for definition of fuzzy concept “near door” as linguistic variable [12]. In this case this statement is instance (or sample) for this determination (may be one from many). If we used just one such instance it means that we determined exact value of variable.

Module is statement in braces. For example, {#X_table := 100; #Y_table := 200;} means determination of coordinates of point in environment

associating with object “table”. These variables are appearing during programming (learning) of robot. May be another statements {#X := 100; #Y :=200;}. In this case we use internal variables embedded in control system of robot and its relation with any object (e.g. “table”) is determined by appearance of corresponding object in context (more “fresh” or “close” allusion of object is used for determination of association between name of object and his coordinates).

Language CBLR may include following internal variables:

#X, #Y, #Z, #go, #stop, #turn, #angle, #front, #back, #left, #right, #fast, #slow, #pick, #break, #on, #off, #object, #color, #manipulate, #up, #down, #top, #bottom, #edge, #center.

These context variables may be oriented on 1) actions (e.g., #go, #stop, #turn, #manipulate, #fast, #slow, #pick, #break), 2) recognition (e.g., #object, #color, #edge, #center) and 3) both (e.g., #X, #Y, #Z, #left, #right, #angle).

During learning we can use following sentences in natural language with introducing of fragments of CBLR, e.g.:

- 1) Go – {#go},
- 2) Table – {#object},
- 3) Go to table – {#object; #go},
- 4) Table is furniture,
- 5) Furniture: table, chair, cupboard,
- 6) Go out – {#go; #object=door; #open; #place=outside; #go},
- 7) Take an apple – {#go; #object=apple; #pick}.

Signs “-”, “;” or word “is” are used for making of association between word or collocation in natural language and one or more declarative description of context variables. Order of these descriptions may be not corresponding to order of employ of them for execution of commands. Figure 4 shows the fragment of knowledge base as result of learning.

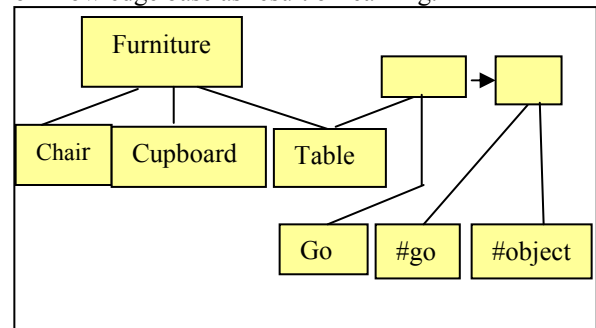


Fig. 4. Fragment of knowledge base with association between collocation (“go”, “table”) and set of two context variables (#go, #object).

On the other hand we can use procedural description (similar to simpler version of Pascal) of any context variables, for example:

1) {#go} – {if place:=find(#place) then go(place) else

if place:=find(#object) then go(place) else say(“Where I must go to?”); }.

2) {#pick} – {object := find(#object); if not near(object) then go(object); if (#object=glass) and not (#opinion=need) then say(“I am afraid to destroy glass”) else pick(#object)}.

In this example we use embedded functions “find”, “go”, “near” and “say” executed by control system of equipment (e.g., robot) and internal variable “place” meaning any coordinates in environment of equipment and variable “object” is a result of recognition of any object. There may be used any other variables with respect to specific aims and features of equipment. When we use any context variable (with sign “#”) during interpretation this variable is replaced by value (string) from context (associated word in natural language). This value (e.g., apple) may be used in vision system as name of any class of recognized objects. What is difference between “#place” and “place”? In first case value of this variable is obtained from context as result of interpretation of language CBLR and in second case value is produced by control system of robot. When control system executes function “find(#place)” or “find(#object)” corresponding argument of this function is replaced by concrete name, e.g. “outside” or “corridor” for #place or “apple” or “hammer” for #object. This replacement may be provided from context and control system finding corresponding object or place by embedded AI based mechanisms such as classical search or deductive methods, neural networks, fuzzy inference, Hidden Markov Model, simple tables and so on. This task includes perception process of control system of intelligent equipment. Function “near(object)” starts recognition process to check the position of robot close or not with any object. Description of object in internal variable “object” which used in control system includes name, coordinates of it and may be another characteristics as result of recognition (finding) of any object from context.

Figure 5 shows an example of interpretation of sentence in natural language as result of usage of associations between words and fragments of CBLR obtained after learning.

Now proposed in this paper approach and language CBLR is implementing (Figure 6) in simulated mobile robot in software used in experiments with robot controlled by neural network.

In this application control system of robot is based on hybrid neural network MLP-ART2 [13]. Task to be

solved by robot is moving to target among obstacles. The dialog based on natural language and learning with proposed in this paper language CBLR provides opportunity to make associations between words-names and obstacles and between words (or phrases) and fragments in CBLR, to program behavior of robot based on internal functions (“go” and “find”) and usage of context variables.

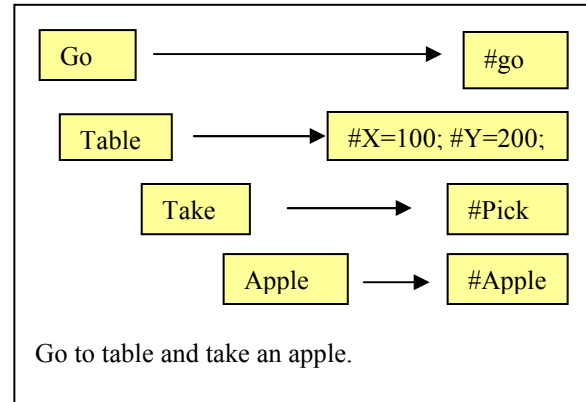


Fig. 5. An example of translation of sentence in natural language to CBLR

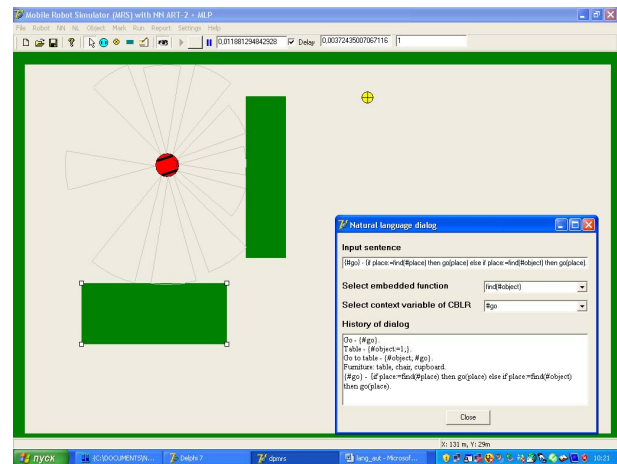


Fig. 6. Screenshot of software for simulation of mobile robot with dialog in natural language and CBLR.

5. Conclusions

In this paper novel approach to programming of intelligent equipment (e.g., robots) is suggested. Feature and advantage of this approach is combination of programming and learning in one system and availability to be expand by introduce in system new components generating and receiving new context. Implementation of this approach as high level language CBLR is proposed.

Unlike known programming languages for robots our language is based on usage of declarative description of context variables and required commands oriented on usage of Natural Language for learning and programming. This approach demands usage of enough smart embedded interpreter generating action as result of interpretation of this description.

Such approach allows to use for learning and programming of robot (and other intelligent equipment) dialog in natural language using context and to achieve easy extendability of abilities of embedded routines for description of more complex behavior of robot.

However this approach can cause difficulties in any industrial applications because sometimes the routines produced during interpretation of CBLR may be unexpected and not appropriate for control of equipment. So the developer must take in account the possibility of this uncertainty and to focus on development of software for careful testing of routines. And it is needed to learn/program intelligent equipment carefully. In this case we expect same difficulties as for development of humanoid robots with Artificial General Intelligence (AGI) [14]. We must solve dilemma between any "freedom" of AGI based robot with easiness of communication with him but uncertain behavior and robot with certain expected behavior but with difficult communication (programming) with him. Solving of this dilemma for industrial applications is ambiguous and demands a special discussion.

Now this language is implementing in simulated mobile robot in software using in experiments with robot controlled by neural network. In future applying of this approach and language CBLR for experiments with real mobile robot is planned.

10. References

- [1] Jun Wan (Editor), *Computational Intelligence in Manufacturing Handbook*, CRC Press LLC, 2001.
- [2] I. Pembeci and G. Hager, "A comparative review of robot programming language", Technical report, CIRL Lab, 2002.
- [3] J.-P. Meynard, "Control of industrial robots through high-level task programming", Thesis, Linköping University, Sweden, 2000.
- [4] B.C. Williams, M.D. Ingham, S.H. Chung and P.H. Elliott, "Model-based programming of intelligent embedded systems and robotic space explorers", *Proceedings of IEEE*, vol. 91, Issue 1, 2003, pp. 212-237.
- [5] P. Hudak, A. Courtney, H. Nilsson and J. Peterson, "Arrows, Robots, and Functional Reactive Programming", *LNCS*, Springer-Verlag, vol. 2638, 2002, pp. 159-187.
- [6] F. Vajda, T. Urbancsek, "High-Level Object-Oriented Program Language for Mobile Microrobot Control", *IEEE Proceedings of the conf. INES 2003*, Assiut - Luxor, Egypt, 2003.
- [7] F. M. Wahl and U. Thomas, "Robot Programming - From Simple Moves to Complex Robot Tasks", *Proc. of First Int. Colloquium "Collaborative Research Center 562 - Robotic Systems for Modelling and Assembly"*, Braunschweig, Germany, 2002, pp. 249-259.
- [8] A.V. Gavrilov, "Dialog system for preparing of programs for robot", *Automatyka*, vol.99, Gliwice, Poland, 1988, pp.173-180 (in Russian).
- [9] A.V. Gavrilov, *Hybrid Intelligent Systems*, NSTU, Novosibirsk, 2003. (in Russian)
- [10] A.V. Gavrilov, "A combination of Neural and Semantic Networks in Natural Language Processing", *Proceedings of 7th Korea-Russia Int. Symp. KORUS-2003*, vol. 2, Ulsan, Korea, 2003, pp. 143-147.
- [11] Kh. T.-Ul Islam, J. Sarkar, K. Hasan, M R. Huq, A.V. Gavrilov, Y.-K. Lee and S.-Y. Lee, "A framework for smart object and its collaboration in smart environment", *Proc. of IEEE ICACT-08*, Korea, February, 2008, pp. 852-855.
- [12] L.A. Zadeh, *The Concept of a linguistic variable and its application to approximate reasoning*, Elsevier Pub. Co., N.Y., 1973.
- [13] A.V. Gavrilov and S.-Y. Lee, "An Architecture of Hybrid Neural Network based Navigation System for Mobile Robot", *Proceedings of 7th International Conference on Intelligent Systems Design and Applications ISDA'07*, Rio-de-Janeiro, Brasil, 22-24 October, 2007, pp. 587-590.
- [14] B. Goertzel and C. Pennachin, *Artificial General Intelligence*, Springer-Verlag, Berlin, Heidelberg, 2007.