New Paradigm of Context based Programming-Learning of Intelligent Agent

Andrey V. Gavrilov[1]

[1]Novosibirsk State Technical University, Kark Marx av., 20,
630092, Russia
Andr_Gavrilov@yahoo.com
http://www.insycom.ru

Abstract. In this paper we propose new paradigm combining concepts "programming", "learning" and "context" for usage in development of control system of mobile robots and other intelligent agents in smart environment. And we suggest architecture of robot's control system based on context and learning with natural language dialog. The core of this architecture is associative memory for cross-modal learning. Main features and requirements for associative memory and dialog subsystem are formulated.

1 Introduction

One of challenges in development of intelligent robots and other intelligent agents is human-robot interface. Two kinds of such known interfaces are 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 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], [8] and now we have a tendency to focus on second kind of languages. But in current time basically 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 future reduction of this gap between manufacturing and service robotics.

On the other hand in service and especially domestic robotics most users are naive about computer language and thus cannot personalize robots using standard programming methods. So at last time robot-human interface tends to usage of natural language [9], [10] and, in particular, spoken language [11], [12]. The mobile robot for example must understand such phrases as "Bring me cup a of tea", "Close the door", “Switch on the light”, “Where is my favorite book? Give it to me”, “When must I take my medication?”.

Using natural language dialog with mobile robot we have to link words and phrases with process and results of perception of robot by neural networks. In [13] to solve this problem the extension of robot programming language by introducing of corresponding operators was proposed. But it seams that such approach is not enough perspective.

In this paper we suggest novel bio- and psychology-inspired approach combining programming and learning with perception of robot based on usage of neural networks and context as result of recognition and concepts obtained by learning during dialog in natural language. In this approach we do not distinguish learning and programming and combine: a declarative (description of context) and procedural knowledge (routines for processing of context implementing elementary behavior) on the one hand, learning in neural networks and ordering of behavior by dialog in natural language on the other hand.

2 Proposed paradigm and architecture of intelligent agent

Our paradigm is based on relationships with other known paradigms and is shown in Fig.1.

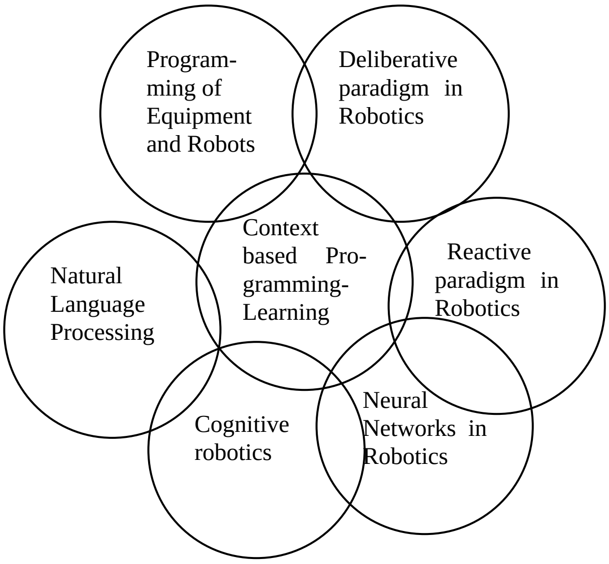

Fig1. Relationships between paradigm “Context based Programming-Learning” and other paradigms.

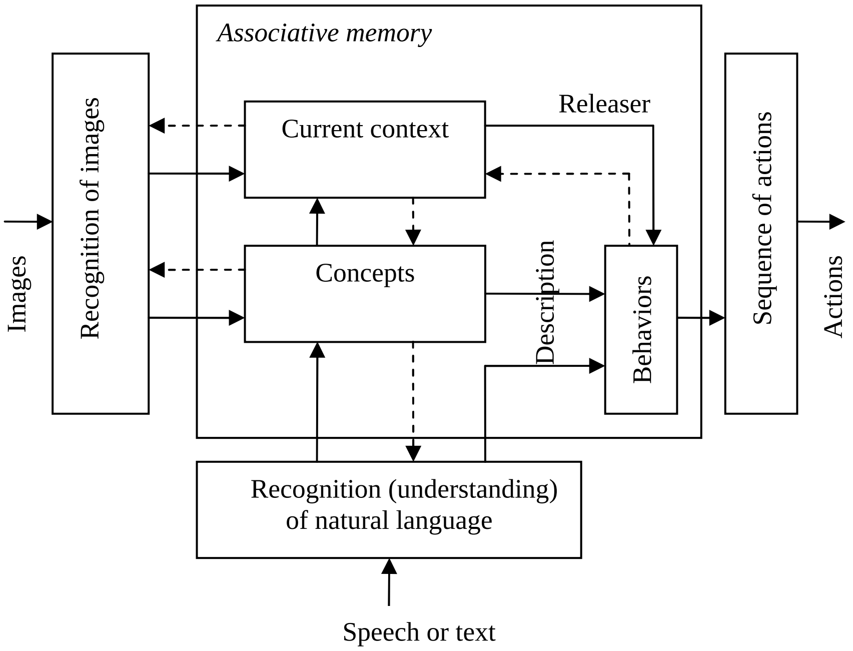

Fig.2. Intelligent agent functional structure oriented on context based programming-learning.

Concepts are associations between images (visual and others) and phrases (words) of natural language. In simple case we will name as concept just name of them (phrases). These phrases are using for determination of context in which robot is perceiving environment (in particular, natural language during dialog) and planning actions. We will name them as *context variables*.

Context is a tree of concepts (context variables) and is recognizing by sensor information and determining by dialog with user. Feedbacks between concepts/context and recognition of images/phrases mean that recognition is controlled by already recognized things.

Dialog with user aims to describe elementary behaviors and conditions for its starting. To start any behavior the system recognizes corresponding concept-releaser as in reactive paradigm of control system of robot.

Behavior is not influence on just actions but also on context. Moreover, behavior may be not connecting directly with actions in environment. In this case we have only thinking under context variables. And even when behavior is oriented on execution of actions it is possible to block this connection and in this case we have any simulation in mind sequence of actions (e.g., it may be mean as planning).

The associative memory must satisfy to follow requirements:

1) to allow usage of both analog and binary inputs/outputs,
2) to provide incremental learning,
3) to provide storing of chain of concepts (as behavior or scenarios).

The elementary behavior is similar to subroutine and contains sequence of actions adaptable to context variables which may be viewed as parameters of subroutine. Of cause, we need to use some primitives directly connecting with elementary actions and describing parameters of these actions or basic context variables. Connections between these primitives and words of natural language must be a prior knowledge of robot obtaining at development of robot or during a preliminary teaching by specialist. It may be simple language similar to CBLR, proposed by author for context based programming of industrial robots in [9], [14]. Feature of this language is absence of different motion primitives. There we have just one motion primitive. All other primitives aims to represent of context variables needed for execution of this motion primitive. To distinguish these primitives and usually using motion or geometric primitives we will call its below as *output primitives*.

3. Cross-Modal Incremental Learning and Associative Memory

Associative memory satisfied to above requirements may be based on hybrid approach and similar to Long-Short Term Memory (LSTM) [15] or table based memory proposed by author in [16].

Concept in Fig. 2 is more common thing as *context*. Concepts are introducing by dialog subsystem for determination of objects, events, properties and abstraction. But *context* consists of several concepts separated by three rules: 1) preliminary defined concepts (names) are using as names of “parameters” to utilize in elementary behavior (see 4); 2) some concepts are using as values of these “parameters”; 3) some concepts (releasers) are using as names of any events causing any behavior.

Dialog subsystem must provide robustness to faults in sentences. For it implementation may be used approach proposed by author in [17] and based on semantic networks combining with neural networks algorithms. This approach is oriented on fuzzy recognition of semantics. Dialog subsystem must provide attend the visual recognition subsystem to link word (phrase) with recognizing image. In this case the recognition subsystem create new cluster with center as current feature vector. And couple of this feature vector and word (phrase) is storing in associative memory for concepts. It may be as result of processing of follow sentence “This is table”. In contrast to that case when system processes sentence “Table is place for dinner” the new cluster is not created and associative memory is used for storing of association between words (phrases).

Thus associative memory must be able to store associations between both couple of symbolic information and couple of word (phrase) and feature vector. Besides concept storing in associative memory must have some tags: basic concept (preliminary defined) or no, name or value, current context or no. And every concept in associative memory must be able to link in chain with other ones. The order of concepts in chain may be defined by dialog subsystem.

4. Output primitives for mobile robot

Set of output primitives may be selected by different way. One approach to that is determination of enough complex behaviors, such as “Find determined object”, “Go to determined place" and so on. These actions may be named as *motion primitives*. Determined *object* and *place* must be obtained from context as value of corresponding context variable. And these variables may be viewed as another kind of output primitives: *context primitives*. In this case such actions must include inside any strong intelligence and ones limit capabilities to learn mobile robot. Another approach may be based on very simple *motion primitives* such as "act", which may be just one. If we want to make capability to say anything by robot not only during dialog inside dialog subsystem we need introduce also at least one motion primitive "say". All another *output primitives* are *context primitives* and ones define features of execution of primitive "act". In other words ones are parameters of subroutine "act". Examples of robot's context variables are shown in Table 1

Table 1. Examples of context primitives.

Name of context primitive	Possible value	How this parameter influence on execution of motion primitive
Object	Name of object	May be used in action "say"
Internal state	Good, Bad, Normal	May cause motion to or from *Object*
Direction	Left, Right, Forward, Back	May cause corresponding turn depending on *Internal state*
Person	Name of person	May be used in action "say"
Obstacle distance	Far, Middle, Close	May be used in "act"
Obstacle type	Static, Dynamic	May be used in "act"
Speed	Low, Normal, High	May be used in "act"

5. Conclusions

In this paper we have suggested new paradigm combining programming and learning of mobile robot based on usage of context and dialog in natural language. Architecture of programming/learning of mobile robot was proposed. Some features of associative memory and dialog subsystem were formulated. Now this architecture is developing for simulated mobile robot and then to be implemented in real robot. This approach may be used for programming/learning of smart object in ambient intelligence [18].

References

1. Jun Wan (Editor): Computational Intelligence in Manufacturing Handbook, CRC Press LLC, (2001).
2. Pembeci I., Hager G.: A comparative review of robot programming language. Technical report, CIRL Lab, (2002).
3. Meynard J.-P.: Control of industrial robots through high-level task programming, Thesis, Linkopings University, Sweden, (2000).
4. Williams B.C., Ingham M.D., Chung S.H., Elliott P.H.: Model-based programming of intelligent embedded systems and robotic space explorers. Proceedings of IEEE, 91(1), (2003) 212-237.
5. Hudak P., Courtney A., Nilsson H., Peterson J.: Arrows, Robots, and Functional Reactive Programming. LNCS, Springer-Verlag, vol. 2638, (2002) 159-187.
6. Vajda F., Urbancsek T.: High-Level Object-Oriented Program Language for Mobile Microrobot Control. IEEE Proceedings of the conf. INES 2003, Assiut - Luxor, Egypt, (2003).
7. Wahl F. M., Thomas U.: 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) 249-259.
8. Samaka M.: Robot Task-Level Programming Language and Simulation. In Proc. of World Academy of Science, Engineering and Technology, Vol. 9, November (2005).
9. Gavrilov A.V.: Dialog system for preparing of programs for robot, Automatyka, vol.99, Glivice, Poland, (1988) 173-180 (in Russian).
10. Lauria S., Bugmann G., Kyriacou T, Bos J., Klein E.: Training Personal Robots Using Natural Language Instruction. IEEE Intelligent Systems, 16 (2001) 38-45.
11. Spiliotopoulos D., Androutsopoulos I., Spyropoulos C.D.: Human-Robot Interaction based on Spoken Natural Language Dialogue. In Proceedings of the European Workshop on Service and Humanoid Robots (ServiceRob '2001), Santorini, Greece, 25-27 June (2001).
12. Seabra Lopes L. et al: Towards a Personal Robot with Language Interface. In Proc. of EUROSPEECH’2003, (2003) 2205—2208.
13. Michael Beetz, Alexandra Kirsch, and Armin M¨uller: RPL$_{\mathrm{LEARN}}$: Extending an Autonomous Robot Control Language to Perform Experience-based Learning. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS 2004, (2004) 1022 – 1029.
14. Gavrilov A.V.: Context and Learning based Approach to Programming of Intelligent Equipment. The 8th Int. Conf. on Intelligent Systems Design and Applications ISDA’08. Kaohsiung City, Taiwan, November 26-28, (2008) 578-582.
15. Hochreiter S., Schmidhuber J.: Long Short-Term Memory. Neural Computation, 9(8) (1997) 1735-1780.
16. Gavrilov A.V. et al: Hybrid Neural-based Control System for Mobile Robot. In Proc. of Int Symp. KORUS-2004, Tomsk, Vol. 1, (2004) 31-35.

17 Gavrilov A.V.: A combination of Neural and Semantic Networks in Natural Language Processing. In Proceedings of 7th Korea-Russia Int. Symp. KORUS-2003, vol. 2, Ulsan, Korea, (2003) 143-147.

18. Gavrilov A.V.: Hybrid Rule and Neural Network based Framework for Ubiquitous Computing. The 4th Int. Conf. on Networked Computing and Advanced Information Management NCM2008, Vol. 2, Gyengju, Korea, September 2-4, (2008) 488-492.