

## КОНТЕКСТНО-ОРИЕНТИРОВАННАЯ ГИБРИДНАЯ АРХИТЕКТУРА СИСТЕМЫ УПРАВЛЕНИЯ ИНТЕЛЛЕКТУАЛЬНОГО РОБОТА

*Аннотация:* В работе предлагается архитектура информационной системы робота, ориентированной на обучение робота с использованием диалога на естественном языке. Особенностью предлагаемого подхода является использование контекстных переменных для конкретизации запускаемого действия (поведения).

*Ключевые слова:* робототехника, искусственный интеллект, естественный язык, машинное обучение

## CONTEXT-ORIENTED HYBRID ARCHITECTURE OF CONTROL SYSTEM OF INTELLIGENT ROBOT

A.V.Gavrilov, Associate Professor, andr\_gavrilov@yahoo.com  
Novosibirsk State Technical University, Novosibirsk, Russia

*Abstract:* The paper presents the architecture of robot's information system oriented to learning via natural language dialog. Feature of proposed approach is usage of context variables to concretize starting actions (behaviour).

*Key words:* robotics, artificial intelligence, natural language, machine learning

В последнее время имеет место тенденция разрабатывать интерфейс пользователя с интеллектуальным роботом, основанный на обучении методом показа (демонстрации) и с использованием естественного языка [1, 2] (в частности, речи [3]).

В докладе предлагается концепция гибридной архитектуры (в соответствии с классификацией в [4]) системы управления роботом, основанной на использовании естественного языка для обучения/программирования робота. При этом поведение робота складывается из действий, запускаемых по мере

заполнения необходимого для них контекста в результате распознавания нейронными сетями образов, поступающих с сенсоров, и диалога с пользователем на естественном языке. В этом подходе практически не различаются «обучение» и «программирование», и в одном процессе обучения объединяются декларативные знания (описание контекста) и процедурные знания (порядок выполнения действий, из которых складывается поведение).

На рисунке показана предлагаемая контекстно-ориентированная архитектура системы управления мобильного робота.

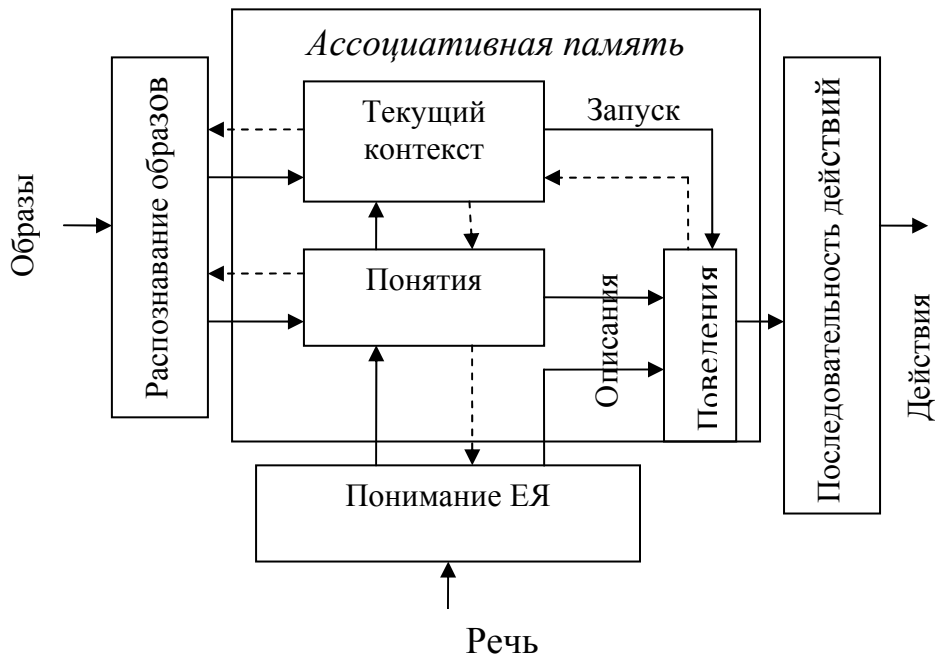


Рис. 1. Контекстно-ориентированная архитектура системы управления мобильного робота

Здесь «понятия» это ассоциации между образами (визуальными и другими) и фразами (словами) естественного языка. В простейшем случае мы будем понимать под понятием просто название образа (слово или словосочетание-фраза). Эти фразы используют для определения контекста, в котором робот воспринимает окружающую среду (в частности естественный язык во время диалога) и планирует действия. Контекст – иерархия понятий (часть из которых используется как параметры действий робота или контекстные переменные), распознаваемых на основе информации с сенсоров и определенных в процессе диалога с пользователем. Обратные связи между понятиями/контекстом и распознаванием образов/фраз означает, что распознавание управляется уже распознанными понятиями.

В процессе обучающего диалога пользователь стремится описывать элементарные поведения и условия для их запуска. Чтобы запустить какое-либо поведение, система распознает соответствующее понятие (т.н. «releaser» в реактивной парадигме системы управления роботом).

Поведение может не быть связано непосредственно с действиями в окружающей среде. В этом случае мы имеем только «размышление» над переменными контекста. И даже когда поведение ориентировано на выполнение действия, эта связь может быть заблокирована, и в этом случае мы имеем дело с моделированием в памяти последовательности действий (например, это может быть в процессе планирования действий).

Ассоциативная память должна удовлетворять следующим требованиям:

- 1) позволять использование аналоговых и бинарных входов/выходов,
- 2) обеспечить продолжающееся (incremental) обучение,
- 3) обеспечить хранение цепочек понятий (как поведений или сценариев).

Элементарное поведение подобно подпрограмме и содержит последовательность действий адаптируемых к переменным контекста, которые могут рассматриваться как аналог параметров подпрограммы. Конечно, мы должны использовать некоторые примитивы, непосредственно соединяющиеся с элементарными действиями и описывающие параметры этих действий, т.е. основные переменные контекста. Связи между этими примитивами и словами естественного языка должны быть предварительные знаниями, полученными роботом на этапе его разработки или настройки во время предварительного обучения робота специалистом. Это может быть простой язык, подобный языку БАЯР [5], предложенному автором для основанного на контексте программирования промышленного транспортного робота (позднее язык CBLR в [6, 7]). Особенность этого языка – наличие всего одного примитива для обозначения движения, использующего для своей детализации значения контекстных переменных. Все другие примитивы предназначены для представления контекстных переменных, необходимых для выполнения этого движения. В таблице приводится примерный набор контекстных переменных для мобильного робота. Кроме примитива *#act*, в набор примитивов, вызывающих реакцию робота, включен примитив *#say*, предназначенный для произнесения фразы в контексте состояния робота и ведущегося им диалога с человеком. Таким образом, все примитивы можно разделить на две группы: контекстные переменные и действия.

Примерные предложения на естественном языке для обучения мобильного робота с использованием последней версии CBLR:

- 1) Идти – *{#act}*.
- 2) Объект – *{#object}*.
- 3) Яблоко, стол – объекты.
- 4) Идти к объекту – *{#object; #act}*.
- 5) Стол это мебель.
- 6) Мебель: стол, стул, книжный шкаф, посудный шкаф.
- 7) Какой объект прямо – *{#direction= Forward; #say}*.
- 8) Найди объект – *{#action=Find; #act}*.

Следует иметь в виду, что эти предложения служат для установления связей между словами ЕЯ и примитивами языка CBLR, а также, другими

Таблица. **Набор контекстных переменных для мобильного робота**

Имя контекстной переменной	Возможные значения	Как этот параметр используется при выполнении примитива действия ( <i>#say</i> или <i>#act</i> )
<i>Action</i>	Имя действия (поведения) или имя подпрограммы, запускаемой примитивом <i>#act</i> , например, Find, Clear, ToGoal и т.п.	Используется для запуска поведений.
<i>Object</i>	Имя объекта	Может быть использован в действии “say”, в поведении Find.
<i>Direction</i>	Left, Right, Forward, Back	Может вызывать соответствующий поворот.
<i>Person</i>	Имя или иной идентификатор одушевленного лица	Может быть использован в действии “say”
<i>Obstacle_Distance</i>	Far, Middle, Close	Может быть использован в действии “act” для обхода препятствий
<i>Obstacle_Type</i>	Static, Dynamic	Может быть использован в действии “act” для обхода препятствий
<i>Speed</i>	Low, Normal, High	Может быть использован в действии “act”
<i>Behavior</i>	Название поведения	Может быть использован в действии “say”
<i>Time</i>	Future, Present, Past	Может быть использован в действии “say”
<i>Place</i>	Имя места	Может быть использован в действии “say”
<i>X, Y</i>	Координаты точки в абсолютной или относительной системе координат (в сантиметрах или метрах, в зависимости от типа робота и задач)	Может быть использован в действии “say” или в “act” для выполнения перемещения
<i>Left, Right, Front, Back</i>	Имя объекта или места, распознаваемое роботом в соответствующем направлении от него	Может быть использован в действии “say” или в “act” для выполнения поворота
<i>Angle</i>	Значение угла относительно направления движения робота	Может быть использован в действии “say” или в “act” для выполнения поворота
<i>Distance</i>	Значение расстояния до некоторой точки, объекта или места (в сантиметрах или метрах, в зависимости от типа робота и задач)	Может быть использован в действии “say”

словами или словосочетаниями. В отличие от них, следующие предложения вызывают действия:

- 1) Иди к столу.
- 2) Какой объект находится прямо?
- 3) Найди яблоко.

В случае работа с манипулятором в набор примитивов-действий можно ввести примитивы #take и #put, обеспечивающие два базовых действия манипулятора – взять и положить. В общем случае, когда мы имеем дело со сложным роботом (например, гуманоидным), который должен обеспечивать сложное поведение, необходимо разбить контекст на иерархию контекстов для каждого вида действия, как показано на рисунке 2.

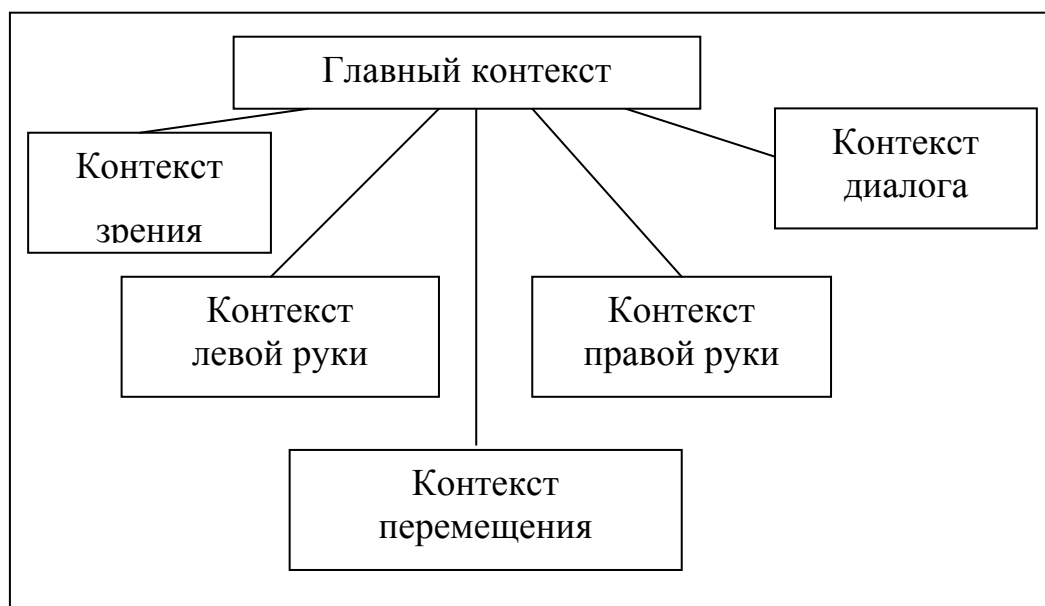


Рис. 2. Иерархия контекстов

При этом каждый вид контекста должен иметь свой независимый набор контекстных переменных и примитивов-действий, т.е., например, при выполнении одноименных примитивов #act для руки и перемещения должны запускаться разные программы. Кроме того, набор контекстных переменных тоже может быть расширен, например, добавлена переменная #Z для третьей координаты. При этом необходимо обеспечить взаимодействие контекстных переменных разных уровней. Например, если контекстная переменная #object неопределена в контексте руки, то ее значение берется из главного контекста, что соответствует нормам объектно-ориентированного программирования. Однако, значение контекстной переменной может и перемещаться вверх по иерархии. Например, контекстные переменные диалога становятся главными контекстными переменными, или распознанный зрительной системой объект (значение переменной #object) становится частью главного

контекста. Также, координаты распознанного объекта, рассчитанные зрительной системой, становятся доступны системой управления рукой (через главный контекст) при выполнении действия #take и могут корректироваться в процессе позиционирования схвата манипулятора.

Для распознавания предложений естественного языка можно использовать метод, предложенный автором в [8] для обучения системы поиска документов и их фрагментов по запросу на естественном языке. Он основан на использовании нейросетевых алгоритмов для распознавания слов и словосочетаний, семантической иерархической сети и упрощенных падежных фреймов.

Обработка команды роботу в виде предложения на ЕЯ состоит из следующих этапов:

- 1) распознавание слов с присваиванием им роли (глагол, существительное, прилагательное или ничего),
- 2) распознавание словосочетаний с использованием иерархической семантической сети (в первую очередь с использованием ролей слов),
- 3) Поиск в семантической сети по ассоциативным связям примитивов или групп примитивов. При этом происходит присваивание значений контекстным переменным.
- 4) Запуск подпрограмм (выполнение притивов типа #act, #say и т.п.).

## Список литературы

1. Lauria S., Bugmann G., Kyriacou T, Bos J., Klein E.: Training Personal Robots Using Natural Language Instruction // IEEE Intelligent Systems – 2001 – N.16 – Pp. 38-45.
2. Spiliotopoulos D., Androutopoulos I., Spyropoulos C.D. Human-Robot Interaction based on Spoken Natural Language Dialogue // Proceedings of European Workshop on Service and Humanoid Robots (ServiceRob '2001), 25-27 June, 2001 - Santorini, Greece, 2001.
3. Seabra Lopes L. et al. Towards a Personal Robot with Language Interface // Proceedings of Int. Conf. EUROSPEECH'2003, 1-4 September, 2003 –Geneva, 2003 – Pp. 2205—2208.
4. R. Murphy. Introduction to AI Robotics / The MIT Press, 2000. – 487p.
5. Гаврилов А.В. Диалоговая система подготовки программ для роботов / А.В.Гаврилов // Glivice, Poland - Automatyka – 1988 - Vol. 99 - Pp. 173-180.
6. Andrey V.Gavrilov. Context and Learning based Approach to Programming of Intelligent Equipment // Proceedings of the 8<sup>th</sup> Int. Conf. on Intelligent Systems Design and Applications ISDA'08, November 26-28, 2008 - Kaohsiung City, Taiwan, 2008 – Pp. 578-582.
7. Andrey V.Gavrilov. New Paradigm of Context based Programming-Learning of Intelligent Agent // Proc. of 1<sup>st</sup> Workshop on Networked embedded and control system technologies. In conjunction with 6<sup>th</sup> International Conference on Informatics in Control, Automation and Robotics ICINCO-2009, 2-5 July, 2009 - Milan, Italy, 2009 – Pp. 94-99.
8. Гаврилов А.В. Архитектура программного обеспечения для поиска документов по запросу на естественном языке // Труды Межд. конф. KDS-2001 "Знание-Диалог-Решение" - С.-Петербург, 2001. - Т.1 - с.124-130.