

## Инструмент наводящего семантического анализа

# 11, ноябрь 2014

Андреев А. М., Гусев А. П.

УДК: 004.891.2

Россия, МГТУ им. Н.Э. Баумана

[arkandreev@gmail.com](mailto:arkandreev@gmail.com)

Развитие современных подходов к реализации интерфейсов управления различными системами и устройствами, подразумевает некоторый уровень интеллектуальности системы (предоставление ограниченного доступа к командам исполнительного устройства и автоматическое выполнение низкоуровневых операций). Таким образом, в системах управления актуально использование новейших методов искусственного интеллекта (ИИ).

Зачастую для этого используют нейронные сети [15], генетические алгоритмы [16], системы логических выводов [17], различные агентные подходы [18].

Существующие интеллектуальные системы (ИС), в том числе и системы искусственного интеллекта, имеют ряд проблем интерпретации существующей базы знаний [15] и корреляции с базой данных:

- Проблема знания системой контекста и программной области;
- Различная форма передачи семантики (смысловой нагрузки);
- Проблема возникновения синонимичных конструкций в базе знаний;
- Самообучения;
- Совместимости новой информации.

Для решения вышеописанных проблем требуется ввести в нашу систему понятие онтогенеза – индивидуального развития организма, как совокупности последовательных морфологических, физиологических и биохимических преобразований, претерпеваемых организмом, от оплодотворения (при половом размножении) или от момента отделения от материнской особи (при бесполом размножении) до конца жизни [19]. Биологическое определение видоизменится при переносе его в другую область, тем самым, определим онтогенез, как процесс самообучения программной оболочки, в виде совокупности последовательных преобразований алгоритмов обработки, и используемых ими данных.

Рассмотрим концепцию универсальной саморазвивающейся системы интерфейса человек-компьютер на основе вызванных потенциалов [20] с использованием видеоаналитики [21], которая сможет адаптироваться к пользователю в процессе работы (т.е. обладает возможностью онтогенеза), с основным ориентиром на достижение качества и удобства управления внешними устройствами.

Концептуальная схема системы представлена на рисунке 1.

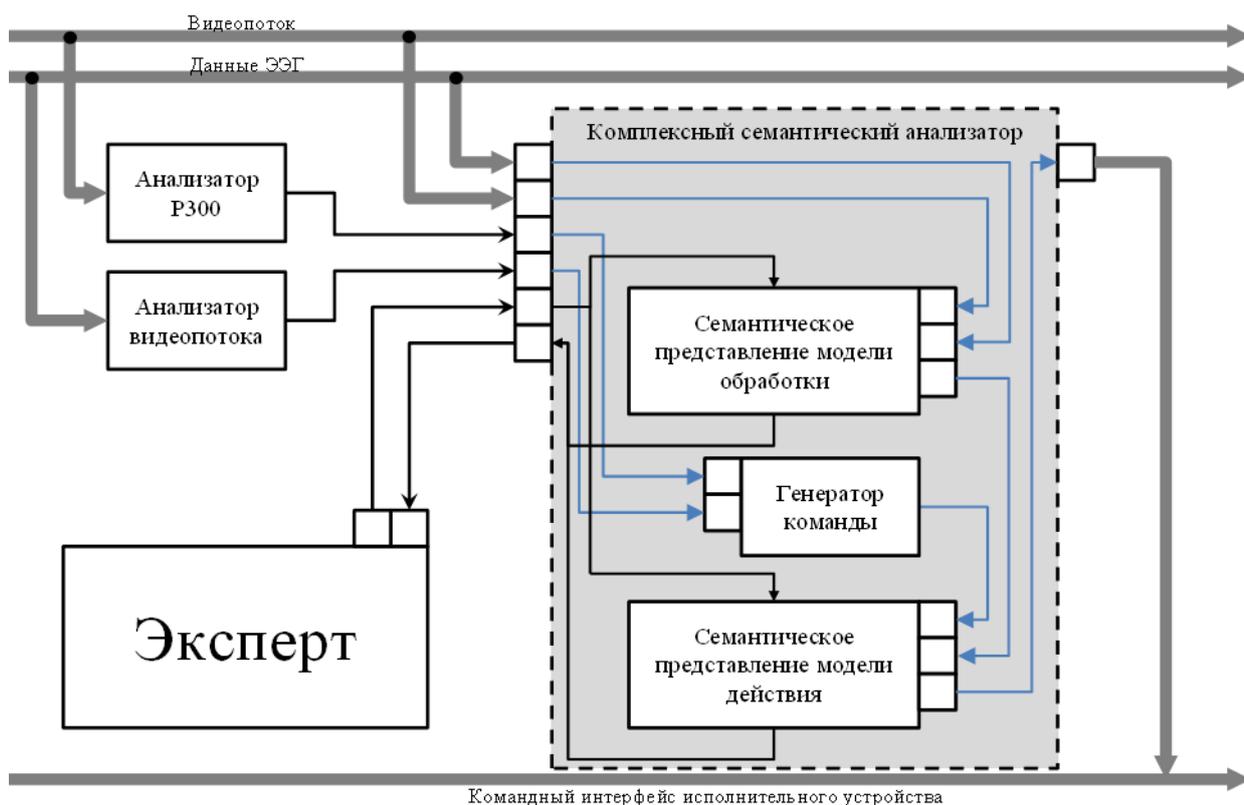


Рисунок 1. Функциональная схема системы в контексте потока данных

Как видно из схемы, для функционирования устройства используются фиксированные алгоритмы детектирования визуальных вызванных потенциалов на электроэнцефалограмме (анализатор Р300) и поиска объектов в видеопотоке (анализатор видеопотока), что является базовым состоянием системы на момент ее запуска (рождения).

Вторичными алгоритмами, являются семантические модели анализа обоих потоков, выполняемые параллельно фиксированным. Такие модели строятся и обучаются для достижения наилучших результатов по некоторому критерию.

Реализация семантики приведена в соответствие с психологией мышления человека, а именно:

- Сравнением приведенных объектов [4];
- Абстрагированием (отвлечением одного свойства объекта от других его свойств) [3];
- Обобщением (объединением похожих свойств разных объектов) [3];
- Конкретизацией (объединением свойства объекта с объектом) [3].

Операциями позволяющими проводить более глобальные операции анализа и синтеза.

Выведем несколько правил внутренней математики нашей системы. Допустим, система имеет свой уникальный алфавит, каждый знак которого будет закодирован целым

беззнаковым числом  $a$ . Используем структуру в виде иерархической семантической сети, имеющей два типа отношений: меронимии (отношение между множеством и подмножеством) и гипонимии (отношение между целым и частями). При этом все элементы в отношении гипонимии, должны подчиняться выработанным правилам.

Поиск конкретизаций (правило,

$$\forall A \forall B, \exists A \subset B: (\forall a_{i+k} = b_i / a_{i+k} \in A, b_i \in B, k \in D, i = \overline{0, |B|})$$



Рис 2. Представление поиска конкретизаций

Поиск эквивалентов (правило,  $\forall A \forall B, \exists A = B: (\forall a_i = b_i / a_i \in A, b_i \in B, i = \overline{0, |B|})$ )



Рис 3. Представление поиска эквивалентов

Поиск обобщений (правило,

$$\forall A \forall B, \exists A \cap B: (\forall a_i = b_i / a_i \in A, b_i \in B, i = \overline{n, m}: n < m, n > 0, m < |B|)$$

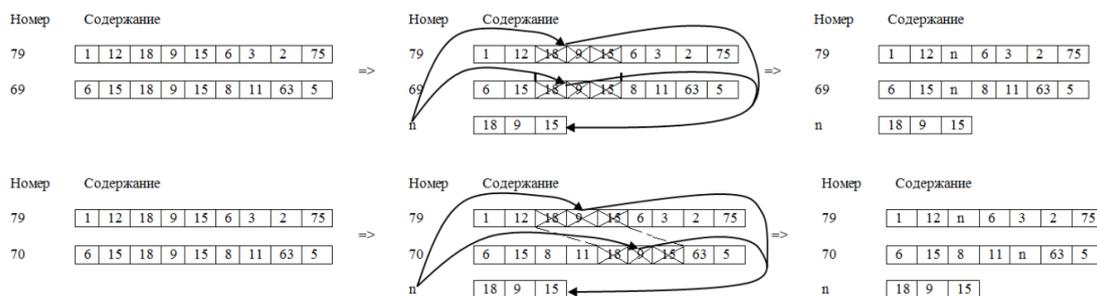


Рис 4. Представление поиска обобщений

Поиск абстракций (правило,

$$\forall A \forall B, \exists A \approx B: (\forall a_i = b_i / a_i \in A, b_i \in B, i = \overline{0, k \& k, |B|}: 0 < k < |B|)$$

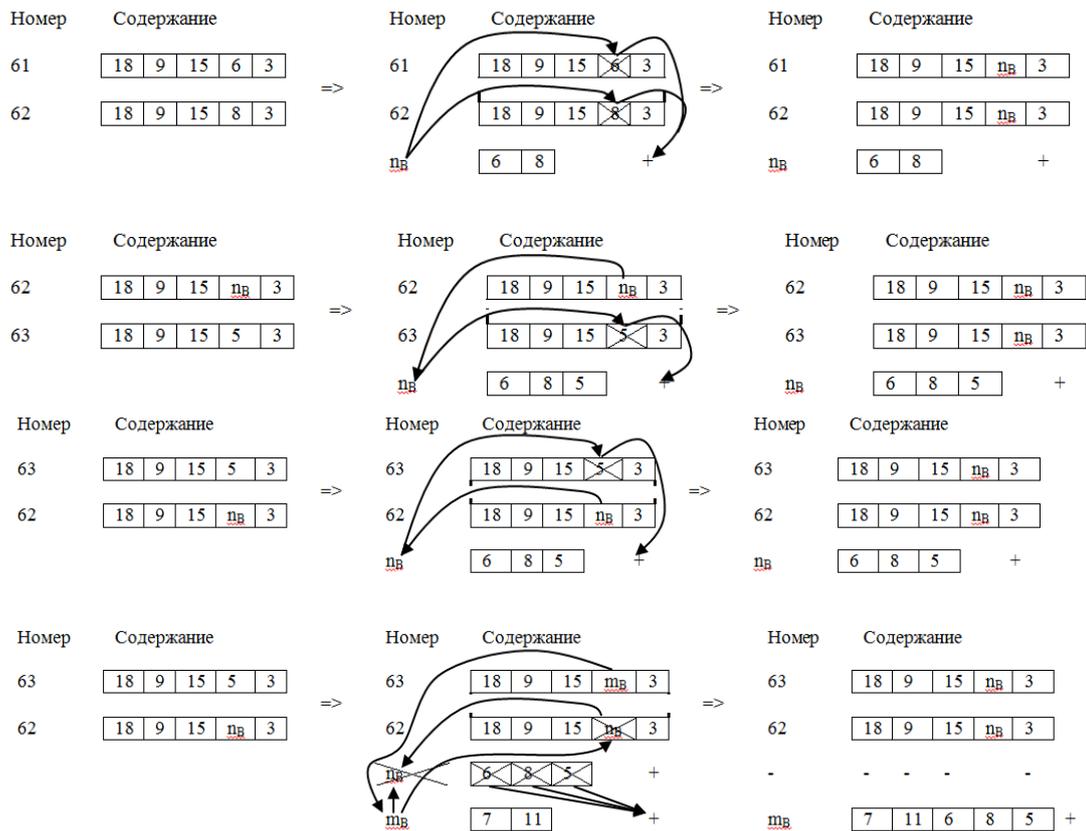


Рис 5. Представление поиска абстракций

При дальнейшей формализации, на основе алгебры кортежей[12], становится ясно, что все эти действия являются попыткой оптимизации структуры[11], за счет исключения определенных типов пересечений. Для этого представим каждый набор элементов в качестве C-кортежа с уникальным целочисленным идентификатором.

C-кортеж, состоит из множеств  $A_i$ , называемых компонентами C-кортежа, и представляет собой их декартово произведение. Его решение представляет собой множество элементарных кортежей:

$$A_1 = \{a, b\}, A_2 = \{c, d\}$$

$$[A_1 \ A_2] = \{(a \ c) \ (a \ d) \ (b \ c) \ (b \ d)\}$$

В нашем случае, составляющие множества содержат единственный уникальный модификатор, который при решении элемента заменяется в зависимости от его типа (type 0 – отношение меронимии, type 1 – гипонимии):

$$Q < type 0 > = (A_1 \ A_2 \ \dots \ A_k)$$

$$Q < type 1 > = \{(A_1) \ (A_2) \ (\dots) \ (A_k)\}$$

Тогда в соответствии с преобразованиями алгебры кортежей, наши правила примут несколько иной вид.

$$\text{Поиск конкретизаций } [B_1 \ B_2 \ \dots \ B_k] \cap \begin{bmatrix} C_1 & C_2 & \dots & C_n & * & \dots & *k \\ * & \dots & *k & C_1 & C_2 & \dots & C_n \end{bmatrix} = \emptyset \quad (1.1)$$

$$\text{Поиск эквивалентов } [B_1 \ B_2 \ \dots \ B_k] \cap [C_1 \ C_2 \ \dots \ C_k] = \emptyset \quad (1.2)$$

$$\text{Поиск обобщений } \begin{bmatrix} B_1 & B_2 \\ \dots & \dots \\ B_{k-1} & B_k \end{bmatrix} \cap \begin{bmatrix} C_1 & C_2 \\ \dots & \dots \\ C_{n-1} & C_n \end{bmatrix} = \emptyset \quad (1.3)$$

$$\text{Поиск абстракций } \left\{ \begin{array}{l} [* \dots B_k] \cap [* \dots C_k] \\ \dots \\ [B_1 \dots *_k] \cap [C_1 \dots *_k] \end{array} \right\} = \emptyset \quad (1.4)$$

Эти операции берет на себя алгоритм эксперта (рисунок 1), причем такая запись правил позволяет эффективно использовать многопроцессорные системы. На интерпретатор семантики системы при этом ложатся более масштабные операции:

- Анализ
- Синтез

Рассмотрим их более подробно.

Представление анализа состоит из нескольких последовательных шагов:

1. Полная декомпозиция. Принятое интерпретатором сообщение необходимо преобразовать в формат используемой алгебры кортежей, для поиска в базе знаний.

$$(n_1 \dots n_k) \Rightarrow \{ \{ (n_1) \} \dots \{ (n_k) \} \}$$

2. Композиция понятий. Разрешенное значение элемента также необходимо привести к общей форме для поиска.

$$\{ \{ (n \dots) \dots \} \dots \} \Rightarrow \{ \{ (n) \dots \} \dots \}$$

3. Резонансное отображение. Результатом анализа интерпретатора является поиск пересечения поискового запроса с разрешенным значением элемента, для его ускорения может быть применен целый спектр методов.

$$\{ \{ (n_1) \} \dots \{ (n_k) \} \} \cap \{ \{ (n) \dots \} \dots \} \neq \emptyset \quad (1.5)$$

Реализация синтеза реакции системы на внешний сигнал, состоит из двух шагов:

1. Поиск первого вхождения. Осуществляется поиск идентификатора в связи гипонимического типа.

$$[n_1 \dots n_k] \cap \begin{bmatrix} id & \dots & * \\ \dots & \dots & \dots \\ * & \dots & id_k \end{bmatrix} \neq \emptyset$$

, что можно записать в виде D-кортежа.

$$[n_1 \dots n_k] \cap [id \dots id_k] \neq \emptyset$$

2. Композиция понятия. В найденной связи в качестве основы генерации используется связь с идентификатором, располагающимся вслед за первым вхождением. И осуществляется синтез реакции с учетом резонансного отображения в виде коэффициентов  $p$ .

3. Разрешение и вывод. Полученный С-кортеж, вычисляется до элементарных кортежей и выбранный по максимальному коэффициенту  $p$  элементарный кортеж, выводится в поток данных.

$$[\{(n \dots) \dots\} \dots] = \{(n \dots)_{p1} \dots (\dots)_{pn}\}$$

Для визуального представления базы знаний, внутреннюю структуру интерпретатора можно представить в виде графа (рисунок 6) или С-системы (1.6). Приведем пример:

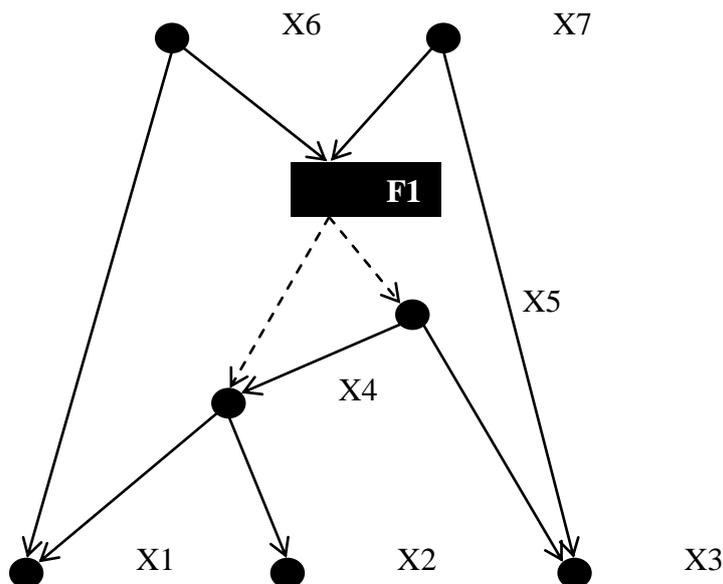


Рисунок 6. Представление связей интерпретатора в виде графа

$$G[XY] = \begin{bmatrix} x_1 & \{\emptyset\} \\ x_2 & \{\emptyset\} \\ x_3 & \{\emptyset\} \\ x_4 & \{x_1, x_2\} \\ x_5 & \{x_4, x_3\} \\ x_6 & \{x_1, F_1\} \\ x_7 & \{F_1, x_3\} \end{bmatrix} \quad (1.6)$$

Условием удачного анализа в интерпретаторе является существование связей вершины  $x$  с вершинами графа транзитивного замыкания  $G^+[XY]$ , содержащимися в кортеже сигнала  $S$ .

Вычисление графа транзитивного замыкания осуществляется по следующей формуле:

$$G^+ = G \cup G^2 \cup \dots \cup G^k, \text{ где } (1.7) \\ k \leq |\log_2(n - 2)| + 2$$

В общем виде, работу контроллера можно представить следующей схемой:

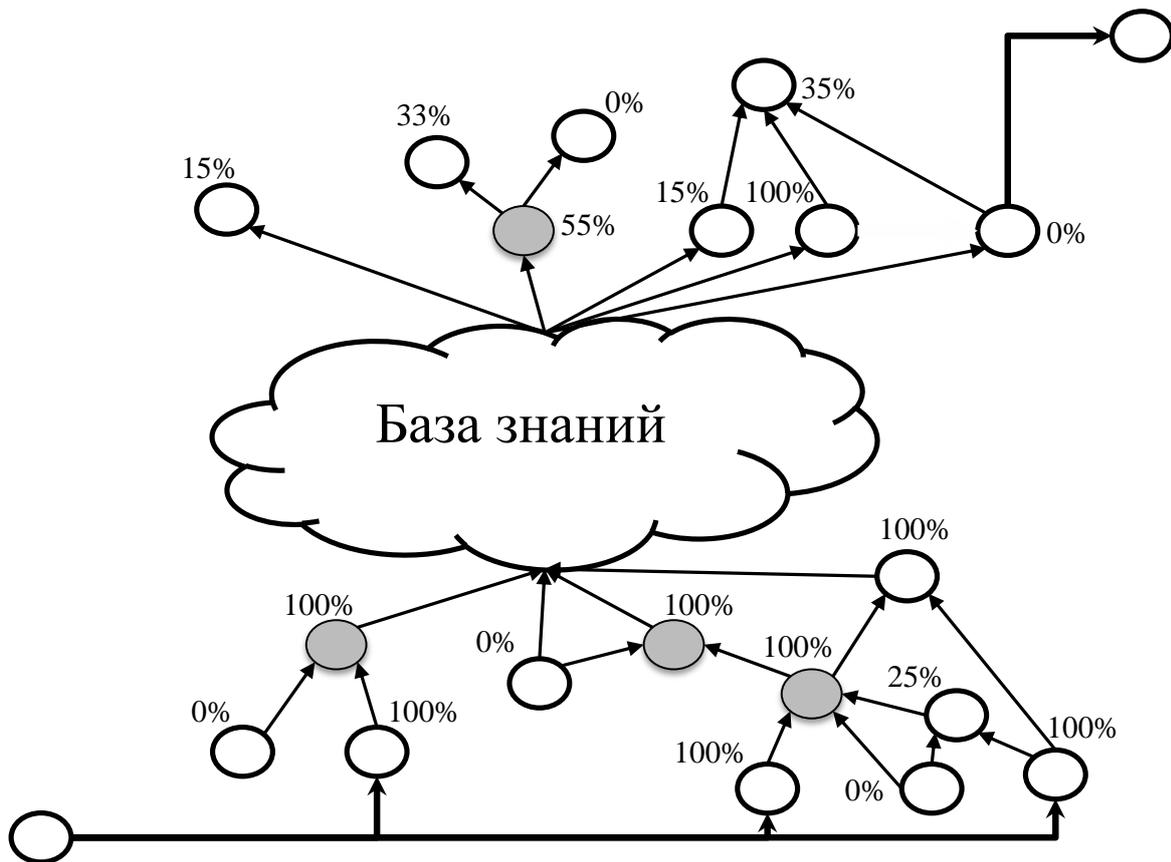


Рисунок 8. Представление связей интерпретатора в виде графа

Такое представление данных, позволяет нам решить ряд проблем, для которых обычно требуются большие затраты времени и вычислительной мощности. К таким проблемам относятся:

- Необходимость разработки алгоритмов сопряжения баз данных и баз знаний, как следствие их различий в теоретических подходах к построению (таблицы и графы в БД, продукции, фреймы, семантические сети в БЗ);
- Трудности распараллеливания при использовании традиционных структур данных;
- Отсутствие простой и прозрачной интерпретации результатов обучения Базы знаний для пользователя.

Дальнейшая разработка системы приводит к генерации фиксированных алгоритмов из полученной семантики и реализации в базовых структурах последующих запусков программы. Такой подход будет гарантировать значительно уменьшенное время запуска системы, упрощенное хранение алгоритмов и существенно уменьшит количество используемой памяти.

## Список литературы

1. К. Куратовский, А. Мостовский Теория множеств / Перевод с английского М. И. Кратко под редакцией А. Д. Тайманова. — М.: Мир, 1970. — 416 с.
2. А. Френкель, И. Бар-Хиллел Основания теории множеств / Перевод с английского Ю. А. Гастева под редакцией А. С. Есенина-Вольпина. — М.: Мир, 1966. — 556 с.
3. Коган А. Б. Нейрофизиологические механизмы мышления человека // Основы физиологии высшей нервной деятельности. — второе, переработанное и дополненное. — Москва: Высшая школа, 1988. — С. 335—350. — 368 с. — 10 000 экз. — ISBN 5-06-001444-4;
4. Маланов, С. В. Психологические механизмы мышления человека: мышление в науке и учебной деятельности / С. В. Маланов — М.: Издательство Московского психолого-социального института; Воронеж: Издательство НПО «МОДЭК», 2004. — 480 с.
5. Тихомиров О. К. Психология мышления. М.: 1984.
6. Dias F.M., Mota A.M. Comparison between Different Control Strategies using eural Networks // 9th Mediterranean Conference on Control and Automation. – Dubrovnik, Croatia, 2001.
7. Venayagamoorthy G.K., Harley R.G., Wunsch D.C. Implementation of Adaptive Criticbased Neurocontrollers for Turbogenerators in a Multimachine Power System”, IEEE Transactions on Neural Networks. – 2003. – Vol. 14, Issue 5. – P. 1047 – 1064.
8. McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity.// Bull. Math. Biophys. – 1943. – v.5. – pp. 115–133.
9. Roussopoulos N.D. A semantic network model of data bases. — TR No 104, Department of Computer Science, University of Toronto, 1976.
10. Quillian, M. R. (1968). Semantic memory. Semantic information processing, 227–270.
11. Кулик Б.А. Математическая модель дедуктивной базы данных на основе алгебры кортежей // Известия РАН. Техническая кибернетика. 1994. №2. С. 161-169.
12. Кулик Б.А. Курс лекции по алгебре кортежей <http://logic-cor.narod.ru>
13. E.W. Sellers and E. Donchin, A P300-based brain-computer interface: initial tests by ALS patients. Clin Neurophysiol: Off J Int Feder Clin Neurophysiol, 117, Mar., 538–548, (2006).
14. B. Blankertz, G. Dornhege, M. Krauledat, K. Müller, and G. Curio, The non-invasive Berlin Brain-Computer Interface: fast acquisition of effective performance in untrained subjects. NeuroImage, 37, Aug., 539–550, (2007).
15. Gallant S.L, Neutral Network Learning and Expert Systems. - Cambridge Mass, MITPress, 1994.
16. Holland J. Concerning efficient adaptive systems // Self-Organizing Systems.- Washington, D.C.: Spartan Books, pp. 215-230, 1962
17. Winston P. H. Artificial Intelligence, 3" ed. Addison Wesley, 1993
18. Bradshaw J. Software Agents. AAAI Press MIT Press, 1997.

19. Gould, S.J. (1977). *Ontogeny and Phylogeny*. Cambridge, Massachusetts: The Belknap Press of Harvard University Press.
20. Sellers, E.W. A P300-based brain-computer interface: initial tests by ALS patients / E.W. Sellers and E. Donchin // *Clin Neurophysiol: Off J Int Feder Clin Neurophysiol*, 117, Mar., 2006. - 538-548p.
21. Richard Szeliski. *Computer Vision: Algorithms and Applications* / Richard Szeliski // Springer, 2010.