АНАЛИЗ СТРУКТУРНЫХ ЗАКОНОМЕРНОСТЕЙ
(Вычислительные системы)
2005 год Выпуск 174

УДК 53.02+519.7+519.812.2

СИСТЕМА УПРАВЛЕНИЯ АНИМАТОМ, ОСНОВАННАЯ НА ТЕОРИИ ФУНКЦИОНАЛЬНЫХ СИСТЕМ П.К.АНОХИНА[1]

А.В. Демин[2], Е.Е. Витяев[3]

Введение

В работах [1-6] была предложена модель работы мозга, формализующая и объясняющая теорию функциональных систем П.К.Анохина. Основываясь на этих работах, в дальнейшем была предложена схема работы функциональных систем [7] и объяснение теории движений Н.А.Бернштейна [8]. В результате была разработана некоторая модель работы мозга. Настало время экспериментальной проверки работоспособности теории.

Цель данной работы состоит в том, чтобы, основываясь на работах [1-9], построить некоторый простейший анимат и показать, что он обучается достаточно эффективно и, по крайней мере, не хуже, чем на основании существующих подходов, основанных на нейронных сетях и потактовом обучении (Reinforcement Learning).

[1]Работа выполнена при финансовой поддержке РФФИ (проект № 05-07-90185в), Интеграционным проектом СО РАН (проект № 119), Программой президента РФ для государственной поддержки научных школ (проект НШ-2112.2003.1).

[2]Институт систем информатики имени А.П. Ершова СО РАН г. Новосибирск

[3]e-mail:vityaev@math.nsc.ru

1. Система управления аниматом

В соответствии с теорией функциональных систем будем считать, что моделируемая система управления аниматом имеет иерархическую архитектуру, в которой базовым элементом системы управления является отдельная функциональная система. При такой архитектуре, функциональные системы верхнего уровня ставят цели системам нижнего уровня. При этом можно считать, что каждая функциональная система решает задачу достижения цели, используя те же методы, что и остальные функциональные системы. На рис.1 приведена архитектура системы управления аниматом ($ФС_j$ — функциональная система).

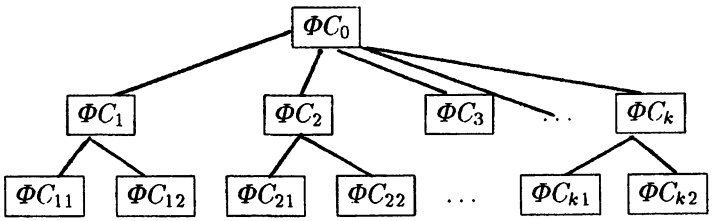

Рис. 1. Архитектура системы управления

Задача отдельной функциональной системы:

- при заданной цели (подцели) и известной информации об окружающей среде и состоянии функциональной системы, найти оптимальный способ достижения цели;
- если на основе прогноза найдено действие, обеспечивающее достижение цели, то дать команду на выполнение этого действия;
- осуществить контроль правильности выполнения действия, т.е. проверить соответствие параметров достигнутого и желаемого результатов.

2. Модель работы функциональной системы

На рис. 2 приведена модель работы функциональной системы, основанная на работе [1]. Будем считать, что в некоторый момент времени функциональной системе ставится цель

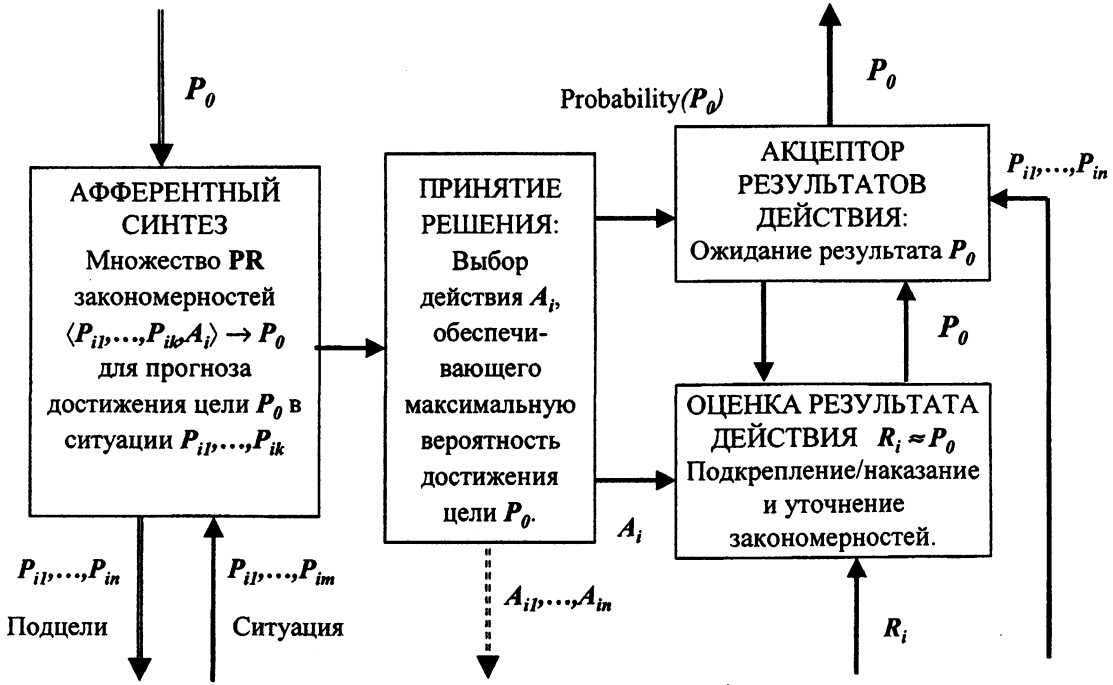

Рис 2. Модель работы функциональной системы

P_0. Цель ставится в виде запроса к функциональной системе — достичь цель P_0. На вход функциональной системы также подается информация об окружающей среде в виде описания ситуации $P_{i1}, \ldots, P_{im}$. В процессе афферентного синтеза из памяти извлекается вся информация, связанная с достижением цели P_0. Эта информация хранится в памяти в виде множества закономерностей вида $\langle P_{i1}, \ldots, P_{ik}, A_i \rangle \rightarrow P_0$. Поскольку информация об окружающей среде уже задана в виде описания ситуации $P_{i1}, \ldots, P_{im}$ еще до постановки цели, то из памяти автоматически извлекается только та информация, связанная с достижением цели, которая может быть применена в данной ситуации. Это достигается использованием (извлечением из памяти) только тех закономерностей, в которых свойства ситуации выполнены, т.е. свойства ситуации P_{ij}, которые есть в условии $P_{i1}, \ldots, P_{ik}$ закономерности должны содержаться в описании ситуации $P_{i1}, \ldots, P_{im}$.

Среди условий $P_{i1}, \ldots, P_{ik}$ закономерности содержатся не только свойства ситуации, но и подрезультаты $P_{i1}, \ldots, P_{in}$, достижение которых необходимо для достижения цели. Достижение подцелей осуществляется отправкой запроса на их достижение вниз по иерархии, что обозначено на рис. 2 двойной стрелкой вниз. Эти запросы активируют в функциональной системе более низкого уровня всю информацию, связанную с достижением этих подцелей. Достижение подцелей может потребовать достижение других подцелей в иерархии целей и т.д. Если какая-то из подцелей не может быть выполнена в данной ситуации (нет закономерностей предсказывающих достижение подцели в данной ситуации), то ответом на запрос будет отказ и соответствующая закономерность исключается из рассмотрения.

Активация закономерностей $\langle P_{i1}, \ldots, P_{ik}, A_i \rangle \rightarrow P_0$ в блоке афферентного синтеза автоматически извлекает из памяти тот набор действий A_i, включая действия, требуемые для достижения подцелей, которые могут привести к достижению цели P_0. Весь этот набор действий вместе с оценками условных вероятностей достижения цели и подцелей передается в блок принятия решений. Блок принятия решений просматривает все действия A_i вместе с активирующими из закономерностями $\langle P_{i1}, \ldots, P_{ik}, A_i \rangle \rightarrow P_0$ и иерархией подцелей и соответствующих дей-

ствий, и выбирает такое действие, которое с учетом вероятностей выполнения подцелей дает максимальную оценку вероятности достижения цели. Далее, действие A_i и все действия, необходимые для достижения подцелей, запускаются на выполнение. В начальной стадии обучения, когда еще нет правил, либо нет ни одного правила, применимого в данной ситуации, действие соответствующей функциональной системы выбирается случайным образом из имеющегося арсенала действий и прогноз отсутствует.

Прогноз ожидаемого результата P_0 и всех подрезультатов для всех подцелей отправляется в акцептор результатов действий. Кроме того, во всех функциональных системах более низкого уровня прогноз подрезультатов также отправляется в акцептор результатов действия соответствующих подсистем.

Данные о полученном результате R_i поступают в акцептор результатов действий блока оценки результата. Проводится сравнение спрогнозированного и полученного результатов. В случае совпадения прогноза и результата, с заданной степенью точности, закономерность, выбранная в блоке принятия решений, подкрепляется, в противном случае, наказывается. Закрепление/наказание состоит в увеличении/уменьшении условной вероятности закономерности. Кроме того, после каждого действия производится уточнение набора правил, как описано в [8]. Если после уточнения для данного состояния находится закономерность с условной вероятностью, большей чем у закономерности, использованной ранее, то новая закономерность будет в дальнейшем использоваться для прогноза и принятия решения.

Систематический вероятностный вывод [10] позволяет найти набор PR закономерностей вида $\langle P_{i1} \,\&\, \ldots \,\&\, P_{ik} \,\&\, A_i\rangle \rightarrow P_0$ с максимальной условной вероятностью, предсказывающий результат P_0 действия A_i в состоянии $\langle P_1, \ldots, P_k\rangle$.

3. Иерархия функциональных систем

Представим функциональные системы более схематично (см. рис. 3) и в качестве иерархии функциональных систем приведем два уровня. Функциональные системы не являются раз

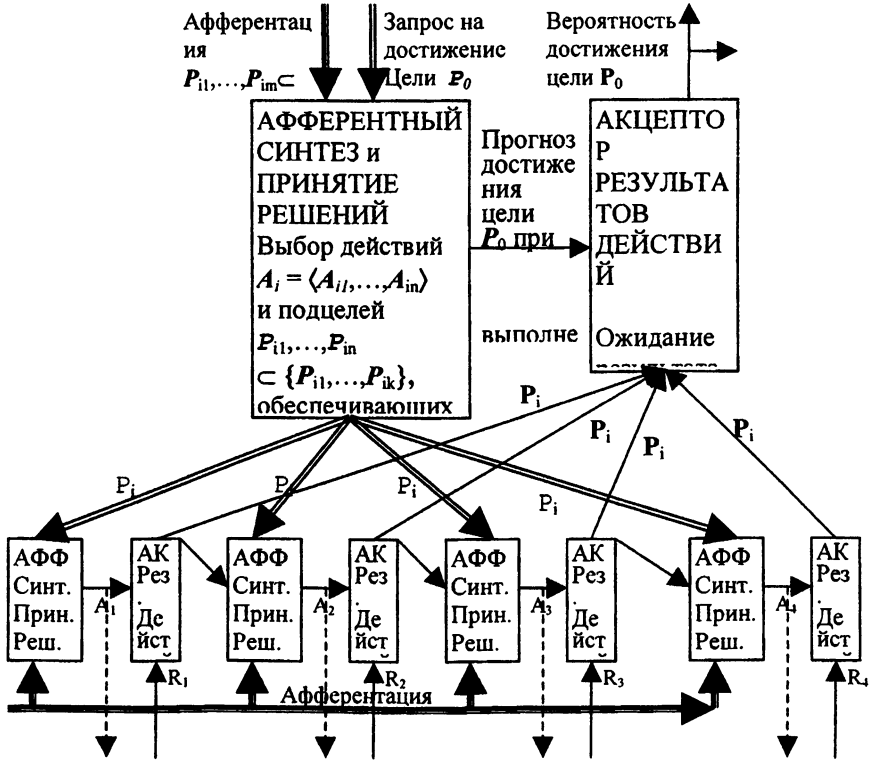

Рис. 3. Два уровня иерархии функциональных систем

и навсегда заданными образованьями. Они меняются и формируются в зависимости от целей. Цели и подцели в свою очередь тоже формируются в зависимости от успешности достижения конечных целей. Покажем, как с помощью закономерностей могут автоматически формироваться цели и подцели.

Расширим понятие результата так, чтобы он мог автоматически формироваться в процессе целенаправленных действий в сложной вероятностной среде:

а) результат должен обладать свойством ветвления: если получен некоторый результат, то дальнейшие действия могут определяться неоднозначно;

б) результат должен содержать набор признаков, которые определяют, что цель цепочки действий достигнута и можно переходить к одной из следующих цепочек действий, т.е. результат — это фиксация законченности действия, обеспечивающая возможность осуществления некоторого следующего действия.

Условие "а" определения результата естественным образом улавливается закономерностями, так как закономерности хорошо прогнозируют результат последовательности некоторых элементарных действий (данного уровня), если эта последовательность действий «стандартна» (начавшись, она продолжается до некоторого результата без изменений). В этом случае с большой вероятностью закономерности прогнозируют выполнение цепочки действий до получения результата. На рис. 3 это действия A_1, A_2, A_3, A_4, приводящие к результатам R_1, R_2, R_3, R_4. Акцептор результатов действия сличает результаты R_1, R_2, R_3, R_4 с предсказанными по закономерностям и, в случае совпадения, выдает ответы P_1, P_2, P_3, P_4 на запросы $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4$. Ответы о достижении цели передаются на входы других блоков (стрелки от акцепторов результатов действий вниз к следующим блокам). Эти ответы автоматически будут включаться в условия закономерностей последующих действий, так как сигнал о том, что предыдущее действие завершено, увеличивает вероятность завершения последующего действия. По определению семантического вероятностного вывода [10] любой сигнал, увеличивающий вероятность прогноза, автоматически включается в условие закономерности.

Условие "б" также выполнено, так как сигналы от обратной афферентации, свидетельствующие о действительном завершении предыдущего действия, увеличивают вероятность достижения результата следующего действия Более детально, на нейронном уровне, процесс автоматического формирования целей и подцелей описан в [4].

4. Описание модели

1. Описание модели. Приведем схему работы анимата, реализующую схему функциональных систем рис. 3 [8]. Будем предполагать, что система управления аниматом функционирует в дискретном времени $t = 0, 1, 2, \ldots$. Пусть анимат имеет некоторый набор сенсоров $S_1, S_2, \ldots, S_n$, характеризующих состояние внешней и внутренней среды, и набор возможных действий $A_1, A_2, \ldots, A_m$. Среди множества сенсоров выделим сенсор SA, который представляет информацию о совершенном действии. Считаем, что история деятельности анимата хранится в таблице данных $X = \{X_1, \ldots, X_t\}$, в которой t-я строка содержит показания сенсоров в момент времени t : $X_t = \{S_1^t, S_2^t, \ldots, S_n^t, SA_t\}, S_1^t, S_2^t, \ldots, S_n^t$ — значения сенсоров $S_1, S_2, \ldots, S_n$ в момент времени t. На множестве X определим множество предикатов $P = \{P_1(t), \ldots, P_k(t), PA_1(t), \ldots, PA_m(t)\}$, где $P_i(t)$ — *сенсорные предикаты*, определяющие некоторые условия на показания сенсоров в момент времени t; $PA_i(t) \Leftrightarrow (SA_t = A_i)$ — *активизирующие предикаты*, показывающие, что в момент времени t было совершено действие A_i.

Введем понятие *предиката-цели* $PG(t) = P_{i1}(t)\ \&\ P_{i2}(t)\ \& \ldots \ldots \&\ P_{il}(t)$, реализующего условие достижения цели в момент времени t.

Каждой функциональной системе $ФС_j$ соответствует некоторая цель G_j, достижение которой является задачей данной функциональной системы и предикат-цель PG_j, характеризующий условие достижения цели.

Каждая $ФС_j$ содержит свой набор предикатов $P_j = P \cup \{PG_{j1}, \ldots, PG_{jn}\}$, где PG_{jk} — предикаты-цели, соответствующие целям нижестоящих по иерархии функциональных систем, подчиненных данной функциональной системе. Каждая

$ФС_j$ содержит множество закономерностей PR_j вида: $\{P_{i1}, \ldots, P_{ik}, PA_i | PG_{j1}, \ldots, PG_{jn}\} \rightarrow PG_j$, где | — означает, что в условии правила стоит только одно из указанных в фигурных скобках выражений. Каждая такая закономерность характеризуется некоторой оценкой p вероятности достижения цели PG_j, при выполнении условия закономерности.

Предположим, что в некоторый момент времени t система $ФС_j$ получила запрос на достижение цели PG_j. Тогда из множества закономерностей PR_j извлекаются все закономерности, условие которых выполнено в текущий момент времени t. Если условие закономерности содержит предикаты-подцели $PG_{j1}, \ldots, PG_{jn}$, то функциональная система отправляет запрос на достижение этих подцелей вниз по иерархии функциональных систем. Среди всех отобранных закономерностей выбирается та закономерность, которая с учетом вероятностей выполнения подцелей дает максимальную оценку f вероятности достижения цели. Оценка f закономерности $\{P_{i1}, \ldots, P_{ik}, PA_i | PG_{j1}, \ldots, PG_{jn}\} \rightarrow PG_j$ вычисляется следующим образом: $f(PG_j | \{PS_{i1}, \ldots, PS_{ik}, PA_i | PG_{j1}, \ldots, PG_{jn}\}) = p \cdot f(PG_{j1}) \cdot \ldots \cdot f(PG_{jn})$, где p — оценка вероятности данной закономерности, $f(PG_{jk})$ оценки вероятностей достижения подцелей. Если все условия выбранной закономерности выполнены, то действие A_i запускается на выполнение. Если множество закономерностей PR_j пусто, либо нет ни одной закономерности, применимой в данной ситуации, то действие выбирается случайно из арсенала имеющихся действий. После совершения действия обновляются показания сенсоров, оценивается результат действия и уточняется набор правил PR_j (см. ниже).

2. Оценка результатов действий. Предположим, что системой $ФС_j$ получен запрос на достижение цели G_j и после достижения цели был получен результат R_j. Определим оценку результата действий. Если после очередного действия предикат цели ложен $PG_j = 0$, то результат не достигнут и оценка результата действий $d^j(t) = 0$. Если результат был получен в момент времени t_0, то все оценки $d^j(t)$, начиная с момента времени t_0 и до предыдущего момента достижения цели t_1, пересчитываются следующим образом: $d^j(t) = re^{\alpha(t-t_0)}, t_1 < t \leq t_0$, где r — функ-

ция оценки качества полученного результата, $\alpha > 0$ — параметр

$$r = \begin{cases} 0, \text{ если } PG_j = 0 \\ \| G_j - R_j \|, \text{ если } PG_j = 1, \end{cases}$$

где $\| \ldots \|$ — мера близости между полученным результатом R_j и поставленной целью G_j. Каждая ΦC_j хранит оценки результатов действий $d^j(t)$ для каждого момента времени t.

3. Генерация правил. Для получения множества закономерностей PR_j, которые использует система ΦC_j, воспользуемся семантическим вероятностным выводом [10].

Семантический вероятностный вывод позволяет находить все закономерности вида $P_{i1}, \ldots, P_{in} \to P_0$ с максимальной вероятностью предсказывающие предикат P_0. Вывод осуществляется на некотором множестве обучающих данных Y с использованием заданного множества предикатов $\{P_1, \ldots, P_m\}$.

Воспользуемся следующим определением вероятностной закономерности, предложенном в работе [12].

Правило $P_{i1}, \ldots, P_{in} \to P_0$ является *закономерностью,* если оно удовлетворяет следующим условиям:

1. $p(P_{i1}, \ldots, P_{in}) > 0$,
2. $\forall \{P_{ij}, \ldots, P_{ik}\} \subset \{P_{i1}, \ldots, P_{in}\}$
   $p(P_0|P_{i1}, \ldots, P_{in}) > p(P_0|P_{ij}, \ldots, P_{ik})$.

Здесь p — оценка условной вероятности правила.

Введем понятие *уточнения* правила. Правило $P_{i1}, \ldots, P_{in}$, $P_{in+1} \to P_0$ является *уточнением* правила $P_{i1}, \ldots, P_{in} \to P_0$, если оно получено добавлением в посылку правила $P_{i1}, \ldots$ $\ldots, P_{in} \to P_0$ произвольного предиката P_{in+1}, и $p(P_0|P_{i1}, \ldots$ $\ldots, P_{in+1}) > p(P_0|P_{i1}, \ldots, P_{in})$.

Алгоритм семантического вероятностного вывода.

• На первом шаге генерируется множество уточнений правила $\to P_0$ (т.е. правила с пустой посылкой). Это множество будет состоять из правил единичной длины, имеющих вид $P_{ij} \to P_0$, для которых $p(P_0|P_{ij}) > p(P_0)$.

• На k-м шаге, $k > 1$, генерируется множество уточнений всех правил, созданных на предыдущем шаге. То есть для каждого правила $P_{ij}, \ldots, P_{ij-1} \to P_0$, сгенерированного на $(k-1)$-м шаге, создается множество правил вида $P_{i1}, \ldots, P_{ik-1}, P_{ik} \to P_0$, таких, что $p(P_0|P_{ij}, \ldots, P_{ij-1}, P_{ik}) > p(P_0|P_{i1}, \ldots, P_{ik-1})$.

• Проверяется нельзя ли из полученных правил удалить какой-то из предикатов так, чтобы при этом условная вероятность правила выросла. Если можно, то такие предикаты удаляются из правила.

• Алгоритм останавливается, когда больше невозможно уточнить ни одно правило.

Для того чтобы избежать генерации статистически незначимых правил, вводится дополнительный критерий — оценка на статистическую значимость. Правила, не удовлетворяющие этому критерию, отсеиваются, даже если они имеют высокую точность на обучающем множестве. Для оценки статистической значимости в алгоритме используется критерий Фишера (точный критерий Фишера для таблицы сопряженности) [13].

Очевидно, что все правила, полученные при помощи данного алгоритма будут являться закономерностями. На рис. 4 представлено дерево вывода, соответствующее описанному процессу.

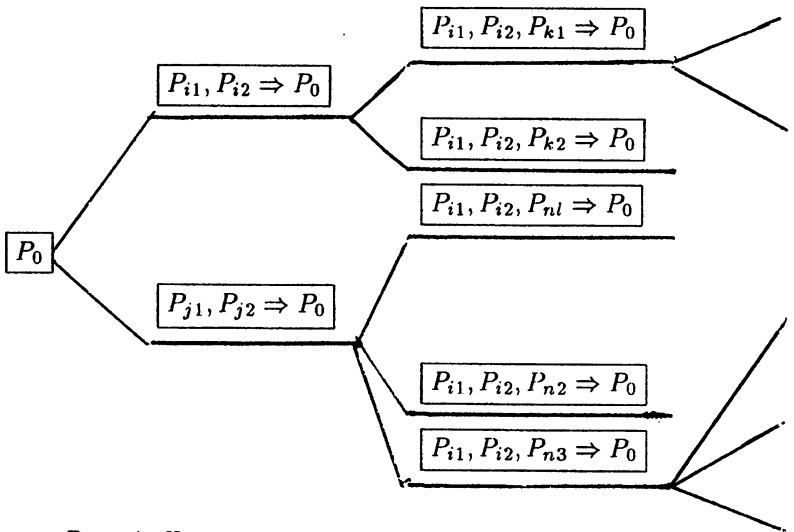

Рис. 4. Дерево семантического вероятностного вывода

Чтобы найти все закономерности $\{P_{i1}, \ldots, P_{ik}, PA_i | PG_{jq}, \ldots$ $\ldots, PG_{jn}\} \rightarrow PG_j$ с максимальной вероятностью предсказывающие достижения цели G_j, строится дерево семантического ве-

роятностного вывода на множестве данных истории деятельности анимата X и множестве оценок действий $d^j(t)$ с использованием набора предикатов P_j, которые использует данная функциональная система. Оценка условной вероятности p правила рассчитывается следующим образом: $p = \frac{\sum\limits_{i \in I} d_i^j}{\|I\|}$, где I — множество моментов времени, когда может быть применено данное правило.

4. Извлечение подцелей. Изначально система управления аниматом имеет заданную априории иерархию функциональных систем. В простейшем случае она может состоять всего из одной функциональной системы. В процессе деятельности система управления может автоматически выявлять новые подцели и порождать соответствующие функциональные системы. Опишем процедуру порождения новых подцелей и функциональных систем.

Предварительно определим два типа подцелей.

1. Подцелями первого типа будем называть ситуации, из которых достижение вышестоящей цели прогнозируется одним правилом, содержащим одну цепочку действий, с высокой вероятностью (вероятность становится близкой к 1).

2. Подцелями второго типа будем называть ситуации, которые увеличивают вероятность достижения вышестоящей цели, но при этом дальнейшие действия могут определяться неоднозначно.

Для выявления подцелей первого типа среди множества правил PR_j функциональных систем выбираются правила вида $R = P_{i1}, \ldots, P_{in}, PA_i \rightarrow PG_j$, имеющие высокую оценку условной вероятности $p > \delta_1$, например, $\delta_1 = 0.9$. Далее, для каждого отобранного правила R порождается подцель G_i и соответствующий предикат-цель $PG_i = P_{i1}$ & P_{i2} &...& P_{in}, равный конъюнкции всех сенсорных предикатов правила R.

Для выявления подцелей второго типа рассматриваются правила с достаточно высокой оценкой условной вероятности $p > \delta_2$ (например, $\delta_2 = 0.7$), имеющие вид $R = P_{il}, \ldots, P_{in}, PA_i \rightarrow PG_j$. Если среди этих правил найдется хотя бы два правила с разными активирующими предикатами, но такими, что все сенсор-

ные предикаты одного правила содержатся в другом, то порождается новая подцель G_i и соответствующий предикат-цель $PG_i = P_{i1}\ \&\ P_{i2}\ \&\ldots\&\ P_{in}$, равный конъюнкции всех сенсорных предикатов, содержащихся в обоих правилах.

Таким образом, порождается новая подцель G_i и соответствующий ей предикат-цель PG_i, если выполнено одно из следующих условий:

1) если существует правило $R = P_{il}, \ldots, P_{in}, PA_i \rightarrow PG_i$, такое, что $p > \delta_1$, то формируется подцель $PG_i = P_{i1}\ \&\ P_{i2}\ \&\ldots\ \ldots\&\ P_{in}$;

2) если существуют правила $R_1 = P_{il}, \ldots, P_{in}, PA_i \rightarrow PG_j$ и $R_2 = P_{jl}, \ldots, P_{jm}, PA_j \rightarrow PG_j$, такие, что $p_1 > \delta_2$, $p_2 > \delta_2$, $\{P_{il}, \ldots, P_{in}\} \subseteq \{P_{jl}, \ldots, P_{jm}\}$ и $A_i \neq A_j$, то формируется подцель $PG_i = P_{i1}\ \&\ P_{i2}\ \&\ldots\&\ P_{in}$.

Затем для каждой выявленной подцели G_i создается новая $ФС_i$, находящаяся ниже по иерархии и реализующая эту подцель. Для созданной системы $ФС_i$ при помощи семантического вероятностного вывода порождается множество закономерностей PR_i. Для этого просматривается все множество данных истории анимата X и выявляются случаи, когда подцель G_i была реализована, и рассчитывается множество оценок действий $d^i(t)$ $ФС_i$ описанным выше способом. Для всех функциональных систем, находящихся на один уровень выше $ФС_i$, набор предикатов обогащается еще одним предикатом PG_i, и генерируются новые правила. Тем самым, множество закономерностей этих функциональных систем обогащается закономерностями, содержащими новую подцель G_i.

5. Эксперимент

1. Описание эксперимента. Для исследования описанной выше системы управления был поставлен следующий простой эксперимент. При помощи компьютерной программы был смоделирован виртуальный мир и анимат, основной целью которого является обнаружение виртуальной «еды». Анимат должен научиться эффективно находить и собирать «еду».

Мир анимата представляет собой прямоугольное поле, разбитое на клетки, и содержащее три типа объектов: «пустые клет-

ки» («трава»), препятствие («препятствие»), и еду («еда»). Объекты «препятствие» располагаются только по периметру виртуального мира, образуя тем самым его естественные границы. Анимат может совершать три типа действий: шагнуть на клетку вперед («шаг»), повернуть налево («налево»), повернуть направо («направо»). Когда анимат шагает на клетку, содержащую «еду», считается, что он ее «поедает», клетка на которой находилась «еда», очищается и новый объект «еда» случайным образом появляется в другом месте поля. Таким образом, количество «еды» в виртуальном мире всегда остается постоянным.

Анимат также обладает девятнадцатью сенсорами, девять из которых информируют его о наличии «еды» («еда на северо–западе», «еда на севере», «еда на северо–востоке», «еда на западе», «еда здесь», «еда на востоке», «еда на юго–западе», «еда на юге», «еда на юго–востоке»), еще девять сенсоров предоставляют информацию о препятствиях («препятствие на северо–западе», «препятствие на севере», «препятствие на северо–востоке», «препятствие на западе», «препятствие на востоке», «препятствие на юго–западе», «препятствие на юге», «препятствие на юго–востоке») и один сенсор говорит о направление анимата («направление»). Сенсор направления показывает ориентацию анимата относительно виртуального мира и может принимать следующие значения: «север», «восток», «юг» и «запад». Сенсоры «еды» и препятствий информируют о наличии данных объектов на клетке, на которой находится анимат, и на соседних с ней клетках, и принимают значения «да» или «нет». Показания этих сенсоров не зависят от ориентации анимата, т.е. сколько бы анимат не крутился на одном месте, их показания не изменятся. Таким образом, чтобы эффективно ориентироваться в виртуальном мире, анимат должен научиться сопоставлять свое направление с положением еды и принимать решение о соответствующем действии.

Изначальный набор предикатов анимата состоит из двадцати двух сенсорных предикатов — по одному на каждый сенсор «еды» и препятствий s : (s=«да»), и четыре на сенсор направления: («направление»=«север»), («направление»=«восток»),

(«направление»=«юг»), («направление»=«запад»), и трех активирующих предикатов: (А=«шаг»), А=«налево»), и (А=«направо»).

Изначально система управления аниматом имеет только одну функциональную систему, целью которой является ощущение наличия «еды» сенсором «еда здесь», соответствующий предикат-цель имеет вида («еда здесь»=«да»). Когда анимат достигает эту цель, то считается, что он «поедает еду».

2. Результаты эксперимента. Одной из основных задач эксперимента была демонстрация возможности автоматического формирования иерархии целей и результатов при целенаправленном поведении. В ходе эксперимента системой управления аниматом были обнаружены подцели, достижение которых значительно увеличивает вероятность достижения цели. Пример первых двух уровней иерархии целей, сформированной аниматом, представлен на рис. 5. На рисунке пунктирной линией

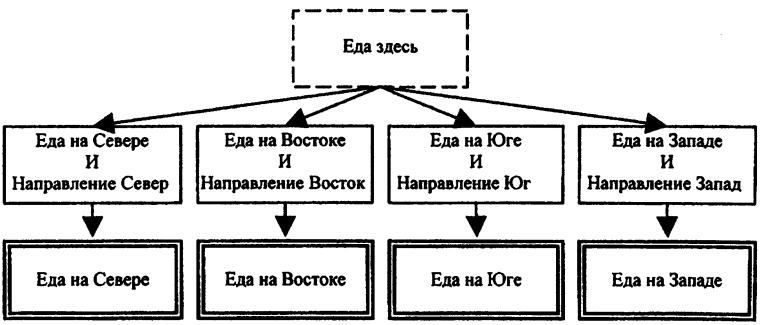

Рис. 5. Иерархия целей, сформированная аниматом в ходе эксперимента

обозначена цель анимата, сплошной линией — подцели первого типа. Подцели, обозначенные двойной линией — это подцели второго типа, поскольку они увеличивают вероятность достижения вышестоящей цели, но при этом дальнейшие действия определяются неоднозначно. К примеру, если после достижения под-

цели «еда на севере» анимат направлен на восток, то ему надо повернуть налево, а если направлен на запад — то направо.

Приведем пример, каким образом анимат использует выработанную им иерархию функциональной системы для достижения цели. Допустим, что анимат ориентирован на восток, и сенсор «еда на северо-востоке» обнаружил «еду». Очевидно, что в данной ситуации, чтобы получить результат, анимат должен совершить три действия: шагнуть вперед, повернуть налево и шагнуть вперед. Рассмотрим, какие цепочки запросов сформирует система управления аниматом, начиная с самой верхней системы:

«еда здесь» $\rightarrow$ *«еда на севере»* & *«направление север»* $\rightarrow$ *«еда на севере»*.

Последняя функциональная система по закономерности *«еда на северо-востоке»* & *«направление»*= *«восток»*& *«действие»*=*«шаг»* $\rightarrow$ *«еда на севере»* с вероятностью 1 запустит на выполнение действие «шаг». Таким образом, будет достигнута подцель «еда на севере». По закономерности *«еда на севере»* & & *«направление»*= *«восток»* & *«действие»*=*«налево»* $\rightarrow$ *«еда на севере»* & *«направление»*=*«север»* с вероятностью 1 анимат повернет налево и достигнет подцель «еда на севере» & «направление север». И, наконец, по закономерности *«еда на севере»* & & *«направление»*=*«север»* & *«действие»*=*«шаг»* $\rightarrow$ *«еда здесь»* анимат шагнет вперед, в результате чего будет достигнута цель, и «еда» будет «съедена».

Для того, чтобы оценить эффективность описанной модели, было решено провести сравнение данной системы управления с системами, построенными на основании теории обучения с подкреплением (Reinforcement Learning), описанной в работах [4].

Для сравнения мы выбрали две системы управления, построенные на основе популярного алгоритма обучения с подкреплением Q–Learning. Суть алгоритма заключается в последовательном уточнении оценок суммарной величины награды $Q(s_t, A_t)$, которую получит система, если в ситуации s_t она выполнит действие A_t по формуле:

$$Q_{i+1}(s_t, A_t) = Q_i(s_t, A_t) + \alpha(r_t + \gamma \max_A(Q_i(s_{t+1}, A) - Q_i(s_t, A_t))).$$

Первая из этих двух систем (Q-Lookup Table) основана на использовании таблицы, которая содержит Q–значения для всех возможных ситуаций и действий. Изначально эти значения таблицы заполняются случайным образом. В процессе работы в каждый такт времени система завершает действие и уточняет соответствующие Q–значения.

Вторая система (Q-Neural Net) использует аппроксимацию функции $Q(s_t, A_t)$ при помощи нейронных сетей. При этом для каждого возможного действия A_i используется своя нейронная сеть NN_i. В каждый такт времени система выбирает действие той нейронной сети, которая выдаёт наибольшую оценку Q–значения, после чего действие совершается и происходит адаптация весов соответствующей нейронной сети по алгоритму Back Propagation.

Для эксперимента было выбрано поле размером 25 на 25 клеток. Весь период функционирования анимата был разбит на этапы по 1000 шагов (тактов). Изначально все системы делают 5000 случайных шагов, чтобы накопить статистику и исследовать окружающую среду, после чего начинают функционировать в обычном режиме. Оценивалось количество «еды», которое соберет анимат с разными системами управления за каждый этап работы. Очевидно, что после того как система управления полностью обучится и достигнет своего оптимального поведения, анимат начнет собирать примерно одно и то же количество «еды» за один этап. Таким образом, можно оценить как эффективность каждой системы управления в целом, так и скорость её обучения.

Было проведено несколько экспериментов, в ходе которых исследовались эффективность и скорость обучения каждой системы управления в случаях с различной плотностью заполнения среды «едой».

Система управления на основе семантического вероятностного вывода во всех случаях показывала более высокую скорость обучения и качество работы по сравнению с системами Reinforcement Learning. Обычно уже после 8,000–10,000 тактов работы система полностью обучалась и достигла своего оптимального поведения.

Система управления на основе нейронных сетей (Q-Neural Net) с использованием отдельной сети для каждого действия не смогла обучиться даже после 100,000 шагов. По этой причине в систему были внесены следующие изменения. Во-первых, было решено использовать одну нейронную сеть для каждого действия и каждого направления анимата. Таким образом, с учетом трех действий и четырех возможных направлений, система использовала 12 нейронных сетей. Во-вторых, задача для системы управления была упрощена путем уменьшения вдвое числа сенсоров, т.е. вместо восемнадцати сенсоров на вход системы подавалось только девять, информирующих о наличии еды. Полученная таким образом система уже смогла научиться решать поставленную задачу.

Нейросетевая система управления оказалась чувствительна к плотности заполнения поля «едой». При большой плотности заполнения система обучается быстрее. Так, при количестве «еды» на поле равным 200, система достигает оптимального поведения в среднем через 30,000–40,000 эпох, тогда как при количестве «еды» равно 100 — через 70,000–80,000. При количестве «еды» равным 50 и меньше система перестает обучаться. Вероятно, это связано с чувствительностью алгоритма Back Propagation к числу показов отдельных примеров — чем меньше «еды» на поле, тем меньшее количество раз нейронной сети предъявляются примеры, связанные с достижением аниматом цели, соответственно, тем меньше влияние этих примеров на процесс обучения. Кроме того, при уменьшении количества «еды» на поле, сильно возрастает нестабильность работы системы. То есть при небольшой плотности заполнения «едой», анимат чаще попадает в ситуации, когда он длительное время не может встретить «еду», в результате чего система начинает «разобучаться».

Система управления с использованием таблицы Q-значений показала плохое качество работы, что объясняется достаточно большим количеством возможных ситуаций: с учетом трех действий и четырех направлений возможно 9984 различных ситуаций. Результаты экспериментов показывают, что даже после 100,000 тактов работы система управления в среднем просматривает только около 4000 ситуаций. Таким образом, даже после

длительного времени обучения возникают ситуации, когда система реагирует неадекватно. Кроме того, система оказалась крайне нестабильной.

Система на основе таблицы значений, также как и система на основе нейронных сетей, оказалась чувствительной по отношению к плотности заполнения среды «едой». Однако, в отличие от нейросетевой системы управления, данная система лучше обучается при небольшой плотности заполнения. Это связано с тем, что при малом количестве «еды» вероятность попасть в ситуацию, в которой анимат уже был ранее, значительно выше, чем при большом количестве «еды». Соответственно, система управления обучается поведению в наиболее вероятных ситуациях. Однако, с уменьшением количества «еды» также возрастает и нестабильность системы. Наилучшие результаты были достигнуты при количестве «еды» равном 100.

На рис. 6 представлены результаты экспериментов для случая, когда количество «еды» на поле поддерживалось равным 100.

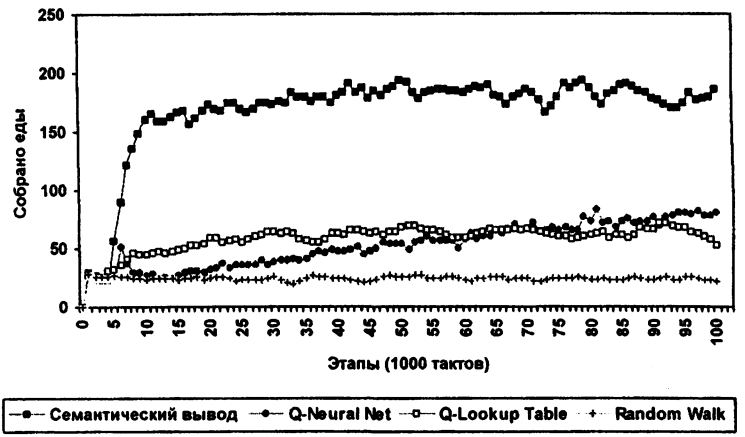

Рис. 6. Количество «еды»,собранное аниматом с разными системами управления

Для каждой системы управления рассчитывались средние значения по результатам двадцати испытаний.

Выводы

В данной модели анимата используется только простейший способ формирования подцелей по сравнению с теми, которые описаны в [8]. Но уже эта возможность дает значительные преимущества в обучении. Поэтому необходимо провести дальнейшие исследования возможности автоматического формирования целей, которые, в конечном итоге, должны дать возможность автоматического построения иерархии целей как это описано в теории движений Н.А.Бернштейна.

Литература

1. ВИТЯЕВ Е.Е. Целеполагание как принцип работы мозга //Модели когнитивных процессов. — Новосибирск, 1997. — Вып.: 158. — Вычислительные системы. — С. 9-52.

2. ВИТЯЕВ Е.Е. Принцип работы мозга и процесс познания в науке и искусстве. Изд. НГУ, Новосибирск, 1995. — 64 с.

3. ВИТЯЕВ Е.Е. Вероятностное прогнозирование и предсказание как принцип работы мозга //Измерение и модели когнитивных процессов. — Новосибирск, 1998. — Вып.: 164. — Вычислительные системы. — С. 14–40.

4. ВИТЯЕВ Е.Е. Формальная модель работы мозга, основанная на принципе предсказания //Модели когнитивных процессов. — Новосибирск, 1998. — Вып.: 164. — Вычислительные системы. — С. 3-61.

5. ВИТЯЕВ Е.Е. Формализация двух принципов работы мозга //Нейроинформатика и ее приложения //Материалы VIII Всероссийского семинара 6–8 октября 2000, Красноярск. — Красноярск, 2000. — С. 33–34.

6. VITYAEV E.E., STARIKOVA I.V. Two new Principles of Brain Activity //PAST CONFERENCES PROCEEDINGS SCI 2002/ISAS 2002 (The 6-th World Multiconference on Systematics, Cybernetics and Information, July 14–18, 2002, Orlando, Florida) VOLUME VIII.

7. МИХЕЕНКО Е.В., ВИТЯЕВ Е.Е. Моделирование работы функциональной системы //VI Всероссийская науч.-техн. конф. "Нейроинформатика-2004". Сб.научных трудов. Ч. 2. — М.: МИФИ, 2004. — С. 124–129.

8. ВИТЯЕВ Е.Е. Объяснение теории движений Н.А.Бернштейна //VII Всероссийская научн.–техн.конф. "Нейроинформатика–2005". Сб. научных трудов. Ч. 1. — М., 2005. — С. 234–240.

9. ВИТЯЕВ Е.Е. Логика работы мозга. Проблемы нейрокибернетики. Т. 2. //Материалы XIV Международной конф. по нейрокибернетике. — Ростов–на–Дону, 2005. — С. 14–17.

10. ВИТЯЕВ Е.Е. Семантический подход к созданию баз знаний. Семантический вероятностный вывод //Новосибирск, 1992. — Вып.: 146. — Вычислительные системы. — С. 19–49.

11. SUTTON R., BARTO A. Reinforcement Learning: An Introduction. — Cambridge: MIT Press, 1998. See also: http://www-anw.cs.umass.tdu/~rich/book/the-book.html

12/ ВИТЯЕВ Е.Е. Метод обнаружения закономерностей и метод предсказания //Эмпирическое предсказание и распознавание образов. — Новосибирск, 1976. — Вып.: 67. Вычислительные системы. — С. 54–68.

13. КЕНДАЛ М., СТЬАРТ А. Статистические выводы и связи. — М.: Наука, 1973. — 899 с.

Поступила в редакцию
10 января 2006 года