

**Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова**

**ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ ЗАДАЧИ ИНФОРМАТИКИ:
НОВЫЕ ПОДХОДЫ И РЕШЕНИЯ**

**Труды школы-конкурса молодых ученых
Новосибирск, Академгородок
25 мая–15 июня 2003 года**

Новосибирск 2003

Сборник содержит статьи, представленные на электронной школе-конкурсе «Новые подходы и решения» аспирантами и молодыми сотрудниками ИСИ СО РАН по следующим направлениям: теоретические аспекты программирования, информационные технологии и информационные системы, системное программное обеспечение, прикладное программное обеспечение.

Программный комитет:

Марчук А. Г. — председатель.

Ануреев И. С. — секретарь.

Касьянов В. Н.

Городня Л. В.

Черноножкин С. К.

Мурзин Ф. А.

Бульонков М. А.

**Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems**

**THEORETICAL AND APPLIED PROBLEMS OF INFORMATICS:
NOVEL APPROACHES AND SOLUTIONS**

**A school-and-contest of young researchers
Novosibirsk, Akademgorodok
May, 25–June, 15, 2003**

Novosibirsk 2003

The volume contains the papers presented by post-graduates and young researchers of A.P. Ershov Institute of Informatics Systems at the school-and-contest «Novel approaches and solutions» which concern the following research areas: theoretical aspects of programming, informational technologies and information systems, system software and applied software.

ПРЕДИСЛОВИЕ

Цель проведения электронной школы-конкурса “Новые подходы и решения” — стимулирование научной деятельности аспирантов и молодых сотрудников (до 35 лет) Института систем информатики СО РАН и их обучение качественному представлению научных работ. Работы принимались в рамках тематики института по следующим направлениям: теоретические аспекты программирования, информационные технологии и информационные системы, системное программное обеспечение, прикладное программное обеспечение. Для участия в конкурсе необходимо было представить работу, включающую описание решаемой задачи, новый подход к решению задачи, сравнение с имеющимися подходами и обоснование новизны. Обучению качественному представлению работ способствовало привлечение к рецензированию самих участников школы-конкурса. При подведении итогов конкурса учитывалось как качество самих работ, так и качество рецензий. Лучшие работы представлены в этом сборнике.

В работе “Трансляция SDL-спецификаций в модифицированные раскрашенные сети Петри” описывается алгоритм автоматического перевода статического подмножества SDL в модификацию раскрашенных сетей Петри — иерархические временные типизированные сети (ИВТ-сети) с последующей генерацией сетевой модели в системе SPV (SDL Protocol Verifier). ИВТ-сети расширяют безопасные раскрашенные сети Петри посредством понятий времени (семантика Мерлина), приоритетов, а также специальных мест, представляющих очереди фишек.

В работе “Коалгебраическое исследование бисимуляционных параллельных процессов” предложено расширение категориального и коалгебраического подходов к описанию поведенческой бисимуляции интерливинговых моделей на модели истинного параллелизма, представленные системами переходов с независимостью и помеченными структурами событий.

В работе “Восстановление положения тела в пространстве по данным, поступающим с одномерных камер для системы реального времени анализа движения объектов” предлагается решение следующей задачи. В пространстве имеется твердое тело с известной геометрией на котором в определенных местах закреплены светоизлучающие диоды, в дальнейшем называемые маркерами, которым присвоены уникальные номера. Также имеется набор одномерных детекторов, расположенных вокруг твердого тела, которые достаточно часто (200–300 раз в секунду) фиксируют положение маркеров. Каждый маркер обладает уникальными характеристиками (цвет, частота мерцания, и т.п.), по которым детектор может отличить его от других мар-

керов и восстановить его номер. В силу одномерности детектора, о каждом маркере, попадающим в его поле видимости, можно получить лишь уравнение плоскости в которой этот маркер содержится. Все плоскости, получаемые от фиксированного детектора, пересекаются по одной прямой (“фокусная прямая”). Задача состоит в том, чтобы по имеющимся данным в реальном времени восстанавливать положение твердого тела с максимальной точностью.

В работе “Статический подход к семантическому структурированию предметных областей для запросов на естественном языке” представлен статистический подход к построению структуры (иерархии) предметной области. Данный подход ориентирован на использование получившейся структуры для поиска информации на естественном языке.

В работе “Теоретико-категорная характеристика языковых эквивалентностей временных параллельных моделей” разработан подход к построению временной частично упорядоченной языковой эквивалентности в контексте моделей реального времени с семантикой “истинного параллелизма” — временных структур событий. Исследуемая эквивалентность формулируется в терминах равенства временных частично-упорядоченных языков системы.

В работе “Построение распределенных объектно-ориентированных интегрированных информационных систем обеспечения безопасности предприятия” рассматривается модель многоуровневой информационной системы, интегрирующей разнородные классы объектов различной конфигурации и степени сложности. Автором предложено архитектурное решение, позволяющее динамически расширять действующую систему, подключая к ней объекты как уже имеющихся в ней классов, так и объекты новых классов. В работе также обобщается опыт практической реализации модели в виде распределенного приложения. Анализируется опыт эксплуатации разработанных систем.

В работе “Программный комплекс, предназначенный для обработки результатов, полученных методом ядерного каротажа нефтяных скважин” обобщен в виде методических рекомендаций опыт применения отечественной технологии для оценки нефте-водонасыщенности коллекторов нефти, разработанной ОАО НПП ВНИИГИС. Изложены физические основы метода, обоснованы аналитические параметры для оценки коэффициентов нефте-водонасыщенности в различных физико-геологических условиях, рассмотрено влияние мешающих факторов, способы качественной и количественной интерпретации данных С/О-каротажа, аппаратура и программное обеспечение метода.

М. Ф. Антонцева

КОАЛГЕБРАИЧЕСКОЕ ИССЛЕДОВАНИЕ БИСИМУЛЯЦИОННЫХ ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОВ

ВВЕДЕНИЕ

Теория категорий в последнее время стала активно использоваться в теории параллелизма для описания и изучения параллельных систем и процессов. Использование методов теории категорий позволило исследовать взаимосвязи между различными моделями [3]. Одним из наиболее широко применяемых понятий этой теории является понятие открытого морфизма.

В теории параллельного программирования существует множество различных поведенческих эквивалентностей, но самыми известными являются бисимуляционные эквивалентности. Две системы называются бисимуляционно эквивалентными, если наблюдатель не может найти различий в их поведении. В рамках теории категорий было предложено абстрактное понятие бисимуляции для различных параллельных моделей, определенное через конструкцию открытых морфизмов. В дальнейшем этот подход стал использоваться и для определения других видов эквивалентностей.

В [4] дан коалгебраический подход для описания поведенческой бисимуляции интерливинговых моделей — систем переходов. Коалгебра является двойственной к алгебре. Двойственность алгебраической и коалгебраической семантик показана в [5].

Цель данной работы — расширить эти два подхода (категориальный и коалгебраический) на модели истинного параллелизма, представленные системами переходов с независимостью и помеченными структурами событий.

1. МОДЕЛИ

1.1. Системы переходов

Определение 1.1.

1. *T-система* — это структура (S, i, A, \rightarrow_S) , где S — множество состояний с начальным состоянием i ; A — множество меток; $\rightarrow_S \subseteq S \times A \times S$ — отношение перехода.

Пишем $s \xrightarrow{a} s'$, если $(s, a, s') \in \rightarrow_S$.

2. **TI-система** — это структура $(S, i, A, \rightarrow_S, I_S)$, где (S, i, A, \rightarrow_S) — T -система и I_S — иррефлексивное, симметричное отношение, такое что

- 1) $(s, a, s') \approx (s, a, s'') \Rightarrow s' = s''$,
- 2) $(s, a, s') I_S (s, b, s'') \Rightarrow \exists u. (s, a, s') I_S (s', b, u), (s, b, s'') I_S (s'', a, u)$,
- 3) $(s, a, s') I_S (s', b, u) \Rightarrow \exists s''. (s, a, s') I_S (s, b, s'')$,
- 4) $(s, a, s') \approx (s'', a, u) I_S (w, b, w') \Rightarrow (s, a, s') I_S (w, b, w')$,

где \approx — это наименьшее отношение эквивалентности, включающее отношение между переходами \prec , которое определяется так:

$$(s, a, s') \prec (s'', a, u) \Leftrightarrow$$

$$\exists b. (s, a, s') I_S (s', b, s''), (s, a, s') I_S (s', b, u), (s, b, s'') I_S (s'', a, u).$$

3. **OPI-система** — это TI -система $(S, i, A, \rightarrow_S, I_S)$, которая является достижимой, ацикличной и такая, что если $s' \xrightarrow{a} s u$, $s'' \xrightarrow{b} s u$ и эти переходы не совпадают, то

$$\exists s \in S. (s, b, s') I_S (s', a, s''), (s, b, s') I_S (s', a, u), (s, a, s'') I_S (s'', b, u).$$

Т.е. в OPI -системе любой квадрат независимости — невырожденный, другими словами, состоит из четырех различных состояний.

Определение 1.2. Пусть $\mathbf{S} = (S, i_S, A_S, \rightarrow_S, I_S)$ и $\mathbf{T} = (T, i_T, A_T, \rightarrow_T, I_T)$ — две OPI -системы. **Морфизм** $f: \mathbf{S} \rightarrow \mathbf{T}$ между ними — это пара $f = (\sigma, \lambda)$, где $\sigma: S \rightarrow T$, $\lambda: A_S \rightarrow A_T$, такие что выполнено:

- 1) $\sigma(i_S) = i_T$;
- 2) если $s \xrightarrow{a} s'$ и $\lambda(a)$ определено, то $\sigma(s) \xrightarrow{\lambda(a)} \sigma(s')$,
если $s \xrightarrow{a} s'$ и $\lambda(a)$ не определено, то $\sigma(s) = \sigma(s')$;
- 3) если $s \xrightarrow{a} s' I_S u \xrightarrow{b} s u'$ и $\lambda(a), \lambda(b)$ определено,
то $\sigma(s) \xrightarrow{\lambda(a)} \sigma(s') I_T \sigma(u) \xrightarrow{\lambda(b)} \sigma(u')$.

OPI -системы вместе с морфизмами формируют категорию, которую обозначают \mathcal{OPI} . Пишем \mathcal{OPI}_A для обозначения категории OPI -систем с множеством меток A .

1.2. Помеченные структуры событий

Определение 1.3. *Помеченная структура событий* — это структура $(E, \leq, \#, l)$, где E — множество событий; $l: E \rightarrow A$ — помечающая функция (A — алфавит); $\leq \subseteq E \times E$ — отношение причинной зависимости; $\# \subseteq E \times E$ — отношение конфликта (иррефлексивное, симметричное отношение), которое удовлетворяет условиям:

- 1) $\forall e \in E. \{d \in E \mid d \leq e\}$ — конечно (так называемый принцип “конечности причин”);
- 2) $\forall e_1, e_2, e_3 \in E. e_1 \# e_2 \leq e_3 \Rightarrow e_1 \# e_3$ — (так называемый принцип “наследования конфликта”).

Два события $e_1, e_2 \in E$ — *параллельны* (пишут $e_1 \text{coe}_2$), если

$$(\neg(e_1 \leq e_2), \neg(e_2 \leq e_1), \{e_1, e_2\} \notin \#).$$

Определение 1.4. Пусть $E = (E, \leq, \#, l)$ — помеченная структура событий. C — *конфигурация*, если выполнено:

- 1) $\forall e, d \in E. e \in C \wedge d < e \Rightarrow d \in C, \preceq \leq id$;
- 2) $\forall e, e' \in E. \neg(e \# e')$.

Множество конфигураций структуры событий обозначается $C(E)$.

Определение 1.5. Пусть $E = (E, \leq, \#, l)$, $E' = (E', \leq', \#, l')$ — две помеченные структуры событий над алфавитами A и A' соответственно. *Морфизм* между ними — это $f = (\eta, \lambda): E \rightarrow E'$, где $\eta: E \rightarrow E'$, $\lambda: A \rightarrow A'$ — частичные функции, такие что выполнено:

- 1) $l' \circ \eta = \lambda \circ l$;
- 2) если c — конфигурация E , то ηc — конфигурация E' , и если $\forall e_1, e_2 \in c$ и их образы определены и $\eta(e_1) = \eta(e_2)$, то $e_1 = e_2$.

\mathbb{E} — категория помеченных структур событий с морфизмами.

2. ОТКРЫТЫЕ МОРФИЗМЫ

Определим категорию мультимножеств $\mathcal{P}om_A$ относительно множества меток A как полную подкатеорию \mathbb{E}_A (категория помеченных структур событий с морфизмами над множеством меток A), чьи объекты — это конечные мультимножества, т.е. частично упорядоченные помеченные события.

Тогда вычисление в структуре событий E можно представить как морфизм в \mathbb{E}_A : $p : P \rightarrow E$, $P \in \mathbb{P} \text{отт}_A$.

Так как структуры событий (и также мультимножества) вкладываются в Π -системы [2, 3], то вычисление в $T \in \mathbb{T}_A$ представляется морфизмом $p : P \rightarrow T$ в \mathbb{T}_A , где P — образ объекта $\mathbb{P} \text{отт}_A$ под действием вложения.

Пусть \mathcal{M} — категория моделей, $\mathbb{P} \text{отт}_A \subseteq \mathcal{M}$ — подкатегория \mathcal{M} . Определим вычисление в $X \in \mathcal{M}$ как морфизм $p : P \rightarrow X$ в \mathcal{M} , где $P \in \mathbb{P} \text{отт}_A$. Тогда любой морфизм в \mathcal{M} $f : X \rightarrow Y$ переводит такое вычисление p в X в вычисление $f \circ p : P \rightarrow Y$ в Y .

Определение 2.1. Пусть $P, Q \in \mathbb{P}$, $X, Y \in \mathcal{M}$. Пусть $f : X \rightarrow Y$ — морфизм в \mathcal{M} , $t : P \rightarrow Q$ — морфизм в \mathbb{P} , $p : P \rightarrow X$ — вычисление в X , $q : Q \rightarrow Y$ — вычисление в Y .

Тогда $f : X \rightarrow Y$ называют **\mathbb{P} -открытым морфизмом**, когда f удовлетворяет условию: если $q \circ t = f \circ p$, то существует морфизм $p' : Q \rightarrow X$, такой что $p' \circ t = p$ и $f \circ p' = q$.

Определение 2.2. Пусть \mathcal{M} — категория моделей, $\mathbb{P} \text{отт}_A \subseteq \mathcal{M}$ — подкатегория \mathcal{M} . Пусть $X_1, X_2 \in \mathcal{M}$. Говорят, что X_1, X_2 — **\mathbb{P} -бисимулятивны** с бисимуляцией X , если существует “хорда” \mathbb{P} -открытых морфизмов $f_1 : X \rightarrow X_1, f_2 : X \rightarrow X_2$.

Предложение 2.3 [2]. *$\mathbb{P} \text{отт}_A$ -открытые морфизмы в \mathbb{T}_A — это морфизмы $(\sigma, 1_A) : T_1 \rightarrow T_2$, где 1_A — тождественное отображение на множестве A , такие что:*

- 1) если $\sigma(s) \xrightarrow{a} T_2 t'$, то $\exists s' \in S. s \xrightarrow{a} T_1 s'$ и $\sigma(s') = t'$;
- 2) если $s \xrightarrow{a} T_1 s', s \xrightarrow{b} T_1 s''$ и $\sigma(s) \xrightarrow{a} T_2 \sigma(s') I_{T_2} \sigma(s') \xrightarrow{b} T_2 \sigma(s'')$, то $s \xrightarrow{a} T_1 s' I_{T_1} s' \xrightarrow{b} T_1 s''$.

3. F-КОАЛГЕБРЫ

Определение 3.1. Пусть $F : \text{Set} \rightarrow \text{Set}$ — функтор. Тогда **F -коалгебра** — это пара (S, α_S) , состоящая из множества S и функтора $\alpha_S : S \rightarrow F(S)$.

Определение 3.2. Пусть (S, α_S) и (T, α_T) — две F -коалгебры. **Гомоморфизм F -коалгебр** (или **F -гомоморфизм**) — это функция $f : S \rightarrow T$, удовлетворяющая равенству $F(f) \circ \alpha_S = \alpha_T \circ f$.

$$\begin{array}{ccc}
 S & \xrightarrow{f} & T \\
 \alpha_S \downarrow & & \downarrow \alpha_T \\
 F(S) & \xrightarrow{F(f)} & F(T)
 \end{array}$$

Композиция двух F -гомоморфизмов тоже F -гомоморфизм, и тождественное отображение F -коалгебр — F -гомоморфизм. Следовательно, набор всех F -коалгебр с F -гомоморфизмами образуют категорию, которая обозначается \mathcal{Set}_F .

Определение 3.3. Пусть (S, α_S) и (T, α_T) — две F -коалгебры. **F -бисимуляция** между ними — это отношение $R \subseteq S \times T$, такое что существует функтор $\alpha_R : R \rightarrow F(R)$ (т.е. (R, α_R) F -коалгебра), удовлетворяющий условию: проекции $\pi_1 : R \rightarrow S$ и $\pi_2 : R \rightarrow T$ являются F -гомоморфизмами.

$$\begin{array}{ccccc}
 S & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & T \\
 \downarrow & & \downarrow \exists \alpha_R & & \downarrow \\
 F(S) & \xleftarrow{F(\pi_1)} & F(R) & \xrightarrow{F(\pi_2)} & F(T)
 \end{array}$$

Теорема 3.4 [4]. Пусть (S, α_S) и (T, α_T) — две F -коалгебры. Функция $f : S \rightarrow T$ является F -гомоморфизмом тогда и только тогда, когда ее граф $G(f)$ — это F -бисимуляция между данными F -коалгебрами.

4. СВЯЗЬ КОАЛГЕБР И МОДЕЛЕЙ

4.1. ОТИ-системы и F^I -коалгебры

Рассмотрим помеченную TI -систему $(S, i_S, A, \rightarrow_S, I_S)$.

Определим $\alpha_S^I : S \rightarrow \mathcal{P}(A \times S \times (S \times A \times S))$ следующим образом:

$$\forall s \in S \quad s \mapsto \{ \langle a, s', I_{as} \rangle \mid s \xrightarrow{a} s' \},$$

где $I_{as} = \{ \langle u, b, u' \rangle \mid s \xrightarrow{a} s' I_S u \xrightarrow{b} s' u' \}$.

Построим $F^I(X) = \alpha_S^I(X \cap S) \quad \forall X \in Set$. Тогда $F^I : Set \rightarrow Set$ — функтор (это легко показать), и (S, α_S^I) — коалгебра. Таким образом, любая TI -система $(S, i_S, A, \rightarrow_S, I_S)$ соответствует F^I -коалгебре. И наоборот, любая F^I -коалгебра соответствует TI -системе $(S, i_S, A, \rightarrow_S, I_S)$:

- $s \xrightarrow{a} s' I_S u \xrightarrow{b} s' u' \Leftrightarrow \langle a, s', u, b, u' \rangle \in \alpha_S^I(s)$;
- $s \xrightarrow{a} s'$ и нет переходов, с ним независимых $\Leftrightarrow \langle a, s', \emptyset \rangle \in \alpha_S^I(s)$.

Т. е. класс TI -систем совпадает с классом F^I -коалгебр. Далее будем рассматривать подкласс TI -систем — $ОТИ$ -системы.

Утверждение 4.1. Пусть $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_T)$ — две $ОТИ$ -системы, (S, α_S^I) и (T, α_T^I) — соответствующие им F^I -коалгебры. Тогда F^I -гомоморфизм между ними — это морфизм $(f, 1_A) : S \rightarrow T$ в $\mathcal{O}TI_A$, который удовлетворяет:

- 1) если $f(s) \xrightarrow{a} t'$, то $\exists s'.$ $s \xrightarrow{a} s'$ и $f(s') = t'$;
- 2) если $s \xrightarrow{a} s'$, $u \xrightarrow{b} s' u'$, $f(s) \xrightarrow{a} f(s') I_T f(u) \xrightarrow{b} f(s') I_T f(u')$, то $s \xrightarrow{a} s' I_S u \xrightarrow{b} s' u'$.

Доказательство. Пусть $f : S \rightarrow T$ — F^I -гомоморфизм между S и T , т.е. удовлетворяет равенству $F^I(f) \circ \alpha_S^I = \alpha_T^I \circ f$. Надо доказать, что для f выполняется:

- 1) если $s \xrightarrow{a} s'$, то $f(s) \xrightarrow{a} f(s')$;

2) если $s \xrightarrow{a} s' I_S u \xrightarrow{b} u'$, то

$$f(s) \xrightarrow{a} f(s') I_T f(u) \xrightarrow{b} f(u');$$

3) если $f(s) \xrightarrow{a} t'$, то $\exists s' \in S$. $s \xrightarrow{a} s'$ и $f(s') = t'$;

4) если $s \xrightarrow{a} s'$, $u \xrightarrow{b} u'$ и $f(s) \xrightarrow{a} f(s') I_T f(u) \xrightarrow{b} f(u')$,
то $s \xrightarrow{a} s' I_S u \xrightarrow{b} u'$.

Пусть $s \xrightarrow{a} s'$, и нет переходов, с ним независимых. Это равносильно тому, что $\langle a, s', \emptyset \rangle \in \alpha_S^I(s)$. Тогда $F^I \langle a, s', \emptyset \rangle = \langle a, f(s'), \emptyset \rangle$ по построению F^I . Так как

$$F^I(f) \circ \alpha_S^I(s) = \alpha_T^I \circ f(s) \quad \forall s \in S,$$

то $\langle a, f(s'), \emptyset \rangle \in \alpha_T^I(f(s))$, что равносильно $f(s) \xrightarrow{a} f(s')$.

Пусть $s \xrightarrow{a} s' I_S u \xrightarrow{b} u'$. Тогда $\langle a, s', u, b, u' \rangle \in \alpha_S^I(s)$ и

$$\langle a, f(s'), f(u), b, f(u') \rangle \in F^I(f) \circ \alpha_S^I(s).$$

Так как f — F^I -гомоморфизм, то $\langle a, f(s'), f(u), b, f(u') \rangle \in \alpha_T^I(f(s))$ и следовательно,

$$f(s) \xrightarrow{a} f(s') I_T f(u) \xrightarrow{b} f(u').$$

Пусть $f(s) \xrightarrow{a} t'$, т.е. $\langle a, t', \emptyset \rangle \in \alpha_T^I(f(s))$. Так как

$$F^I(f) \circ \alpha_S^I(s) = \alpha_T^I \circ f(s) \quad \forall s \in S,$$

то $\langle a, t', \emptyset \rangle \in F^I(f) \circ \alpha_S^I(s)$. Отсюда следует, что существует $s \in S$, такое что $s \xrightarrow{a} s'$ и $\langle a, t', \emptyset \rangle = F^I(f) \langle a, s', \emptyset \rangle = \langle a, f(s'), \emptyset \rangle$. Поэтому $t = f(s')$.

Пусть $s \xrightarrow{a} s'$, $u \xrightarrow{b} u'$ и $f(s) \xrightarrow{a} f(s') I_T f(u) \xrightarrow{b} f(u')$.

Тогда

$$\langle a, f(s'), f(u), b, f(u') \rangle \in \alpha_T^I(f(s)).$$

Так как $F^I(f) \circ \alpha_S^I(s) = \alpha_T^I \circ f(s) \quad \forall s \in S$, то

$$\langle a, f(s'), f(u), b, f(u') \rangle \in F^I(f) \circ \alpha_S^I(s),$$

т.е. $s \xrightarrow{a} s' I_S u \xrightarrow{b} u'$.

Таким образом, мы показали, что если $f: S \rightarrow T$ — F^I -гомоморфизм между S и T , то он является морфизмом, который удовлетворяет пп. 1–2 утверждения.

Обратное очевидно.

Утверждение 4.1 доказано. \blacksquare

Теорема 4.2. Пусть $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_T)$ — две ОП-системы, (S, α_S^I) и (T, α_T^I) — соответствующие им F^I -коалгебры. Тогда F^I -бисимуляция между ними — это отношение $R \subseteq S \times T$, такое что $\forall \langle s, t \rangle \in R$ выполнено:

- 1) $\forall s' \in S$ если $s \xrightarrow{a} s'$, то $\exists t' \in T$. $t \xrightarrow{a} t'$ и $\langle s', t' \rangle \in R$;
- 2) $\forall t' \in T$ если $t \xrightarrow{a} t'$, то $\exists s' \in S$. $s \xrightarrow{a} s'$ и $\langle s', t' \rangle \in R$;
- 3) $\forall s', u, u' \in S$ если $s \xrightarrow{a} s' I_S u \xrightarrow{b} s' u'$, то
 $\exists t', w, w' \in T$. $t \xrightarrow{a} t' I_T w \xrightarrow{b} t' w'$ и $\langle s', t' \rangle, \langle u, w \rangle, \langle u', w' \rangle \in R$;
- 4) $\forall t', w, w' \in T$ если $t \xrightarrow{a} t' I_T w \xrightarrow{b} t' w'$, то
 $\exists s, u, u' \in S$. $s \xrightarrow{a} s' I_S u \xrightarrow{b} s' u'$ и $\langle s', t' \rangle, \langle u, w \rangle, \langle u', w' \rangle \in R$.

Доказательство. Пусть $R \subseteq S \times T$ — F^I -бисимуляция между $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_T)$. Тогда по определению 3.3 существует функтор $\alpha_R^I: R \rightarrow F(R)$ такой, что (R, α_R^I) — F^I -коалгебра. Тогда α_R^I индуцирует отношение перехода $\rightarrow_R \subseteq R \times A \times R$.

Пусть $\langle s, t \rangle \in R$, $s \xrightarrow{a} s'$ и $\pi_1 \langle s, t \rangle = s$. Из этого следует, что $\pi_1 \langle s, t \rangle \xrightarrow{a} s'$, и так как π_1 — F^I -гомоморфизм, то по определению 3.3 существует $\langle s'', t' \rangle \in R$ такое, что $\langle s, t \rangle \xrightarrow{a} \langle s'', t' \rangle$ и $\pi_1 \langle s'', t' \rangle = s'$. Следовательно, $\langle s', t' \rangle \in R$. Так как π_2 — F^I -гомоморфизм, то $t \xrightarrow{a} t'$.

Таким образом мы доказали, что для любого $s' \in S$, если $s \xrightarrow{a} s'$, то $\exists t' \in T$. $t \xrightarrow{a} t'$ и $\langle s', t' \rangle \in R$. Доказательство 2–4 — аналогично.

Мы доказали, что если отношение $R \subseteq S \times T$ — F^I -бисимуляция, то R удовлетворяет 1–4.

Обратно: пусть $R \subseteq S \times T$ удовлетворяет 1–4.

Определим $\alpha'_R : R \rightarrow F(R)$ для $\langle s, t \rangle \in R$ следующим образом:

$$\alpha'_R \langle s, t \rangle = \left\{ \langle \langle a, \langle s', t' \rangle, I_{a, \langle s, t \rangle} \rangle \mid s \xrightarrow{a} s', t \xrightarrow{a} t' \text{ и } \langle s', t' \rangle \in R \right\},$$

$$\text{где } I_{a, \langle s, t \rangle} = \left\{ \begin{array}{l} \langle \langle \langle u, w \rangle, b, \langle u', w' \rangle \rangle \mid s \xrightarrow{a} s' I_S u \xrightarrow{b} s' u', \\ \qquad \qquad \qquad \qquad \qquad \qquad t \xrightarrow{a} t' I_T w \xrightarrow{b} t' w' \\ u \langle s', t' \rangle, \langle u, w \rangle, \langle u', w' \rangle \in R \end{array} \right\}.$$

Тогда легко показать, что (R, α'_R) — F^I -коалгебра и проекции из R в S и в T — это F^I -гомоморфизмы. Следовательно, отношение $R \subseteq S \times T$ — F^I -бисимуляция между $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_S)$.

Теорема 4.2 доказана. ■

Теорема 4.3. *Для ОТИ-систем над алфавитом A $\mathbb{P}om_A$ -открытые морфизмы совпадают с F^I -гомоморфизмами соответствующих им F^I -коалгебр.*

Доказательство. Следствие утверждения 4.1 и предложения 2.3. ■

Теорема 4.4. *Для ОТИ-систем над алфавитом A $\mathbb{P}om_A$ -бисимуляция совпадает с F^I -бисимуляцией соответствующих им F^I -коалгебр.*

Доказательство. Пусть $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_S)$ — две ОТИ-системы, (S, α'_S) и (T, α'_S) — соответствующие им F^I -коалгебры. Пусть $R \subseteq S \times T$ — F^I -бисимуляция между ними. Так как по теореме 4.3 F^I -гомоморфизмы совпадают с $\mathbb{P}om_A$ -открытыми морфизмами, то по определению 2.2 R — $\mathbb{P}om_A$ -бисимуляция между $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_S)$.

Обратно: пусть R — $\mathbb{P}om_A$ -бисимуляция между $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_S)$. По определению 2.2 существуют $\mathbb{P}om_A$ -открытые морфизмы $f_1 : R \rightarrow S$, $f_2 : R \rightarrow T$, которые по теореме 4.3 являются F^I -

гомоморфизмами. В качестве f_1 и f_2 возьмем проекции из R в S и в T . Строим $\alpha_R^I: R \rightarrow F^I(R)$, как в доказательстве теоремы 4.2. Тогда (R, α_R^I) — F^I -коалгебра и проекции из R в S и в T — это F^I -гомоморфизмы. Т. е. отношение $R \subseteq S \times T$ — F^I -бисимуляция между (S, α_S^I) и (T, α_T^I) .

Теорема 4.4 доказана. ■

Теорема 4.5. Пусть $(S, i_S, A, \rightarrow_S, I_S)$ и $(T, i_T, A, \rightarrow_T, I_T)$ — две ОТИ-системы. Функция $f: S \rightarrow T$ является $\mathbb{P}om_A$ -открытым морфизмом тогда и только тогда, когда ее граф $G(f)$ — это F^I -бисимуляция соответствующих им F^I -коалгебр.

Доказательство. Следствие теоремы 3.4, теоремы 4.3 и теоремы 4.4 ■

4.2. Помеченные структуры событий и коалгебры

Рассмотрим помеченную структуру событий $(E, \leq, \#, l)$. Определим ее “развертку” $les.otsi(E)$ как ПИ-систему $les.otsi(E) = (S, i_S, A, \rightarrow_S, I_S)$ [3], где S — множество конечных конфигураций структуры событий, $i_S = \emptyset$;
 $A = l(E)$;

$$c \xrightarrow{a} c' \Leftrightarrow c = c' \setminus \{e\}, \quad l(e) = a, \quad e \in E;$$

$$c \xrightarrow{a} c' I_S \bar{c} \xrightarrow{b} \bar{c}' \Leftrightarrow (c' \setminus c) c o (\bar{c}' \setminus \bar{c}).$$

Утверждение 4.6 [3]. $les.otsi(E)$ — ОТИ-система.

Следующая теорема является следствием теоремы 4.4.

Теорема 4.7. Две помеченные структуры событий $\mathbb{P}om_A$ -бисимулятивны тогда и только тогда, когда F^I -коалгебры, соответствующие их “разверткам” в ОТИ-системы, F^I -бисимулятивны.

ЗАКЛЮЧЕНИЕ

Статья посвящена исследованию распределенных систем, таких как системы переходов с независимостью и помеченные структуры событий. Для исследования применялись два подхода, которые стали активно использо-

ваться в теории параллелизма сравнительно недавно для спецификации и верификации параллельных процессов. Это коалгебраический и категориальный подходы. В ходе работы стало ясно, что некоторые определенные свойства интерливинговых моделей можно распространить на модели с “истинным” параллелизмом, такие как *ОП*-системы. Однако для *П*-систем эти свойства не выполняются.

В дальнейшем планируется, используя коалгебраические методы, изучить взаимосвязи между различными временными сетевыми моделями, а также исследовать новые поведенческие эквивалентности, учитывая также временные аспекты поведения систем.

СПИСОК ЛИТЕРАТУРЫ

1. **Вирбицкайте И. Б.** Семантические модели в теории параллелизма. — Новосибирск: ИСИ СОРАН, 2000. — 190 с.
2. **Joyal A., Nilsen M., Winskel G.** Bisimulation From Open Maps // Information and Computation. — 1996. — N 127. — P. 164–185.
3. **Nilsen M., Sassone V., Winskel G.** Relationships Between Models of Concurrency // Theor. Comput. Sci. — 1994. — N 803. — P. 425–476.
4. **Rutten J. J.** Universal coalgebra: a theory of systems // Theor. Comput. Sci. — 2000. N 249. — P. 3–80.
5. **Rutten J. J., Turi D.** Initial Algebra and Final Coalgebra Semantics for Concurrency // EATS. — 2000. — N 62. — P. 530–559.

Аргиров Виктор Сергеевич

ТРАНСЛЯЦИЯ SDL-СПЕЦИФИКАЦИЙ В МОДИФИЦИРОВАННЫЕ РАСКРАШЕННЫЕ СЕТИ ПЕТРИ

1. ВВЕДЕНИЕ

Верификация распределенных систем, таких как коммуникационные протоколы, — актуальная проблема современного программирования. Для представления распределенных систем часто используется язык выполнимых спецификаций SDL [1, 2], принятый в качестве стандарта ITU. Преимущество SDL в его выразительной силе однако, именно оно и затрудняет анализ и верификацию спецификаций этих систем. Один из подходов состоит в автоматическом переводе спецификаций распределенных систем в модели, для которых разработаны эффективные методы анализа. В качестве моделей выбраны модифицированные раскрашенные сети Петри, названные иерархическими временными типизированными сетями (ИВТ-сетями) [3]. ИВТ-сети расширяют безопасные раскрашенные сети Петри посредством понятий времени (семантика Мерлина), приоритетов, а также специальных мест, представляющих очереди фишек.

В данной работе описывается автоматический перевод статического подмножества SDL и генерация сетевой модели в системе SPV (SDL Protocol Verifier). Эта работа частично поддержана грантом РФФИ № 03-07-90331В.

2. СИСТЕМА SPV

Система SPV представляет собой интегрированный программный комплекс для проектирования, анализа и симуляции моделей ИВТ-сетей.

Комплекс SPV включает:

- графический редактор иерархических сетевых моделей;
- транслятор с языка SDL;
- симулятор — блок имитационного моделирования и отладки.

Типичный цикл работы пользователя в системе SPV выглядит следующим образом. Строится исходная сетевая модель исследуемой системы, что может быть сделано непосредственно в графическом редакторе системы

SPV, либо эта сетевая модель получается каким-либо другим способом, например, как результат трансляции с языка выполнимых спецификаций SDL. Затем проводится симуляция модели в ручном или автоматическом режиме. Система позволяет визуально контролировать ход симуляции, отслеживать изменение данных, устанавливать контрольные точки останова и т. п. По результатам симуляции исходная модель уточняется, и цикл разработки повторяется до получения удовлетворительных результатов.

Симулятор позволяет проследить за процессом функционирования сети при заданной начальной разметке. При этом часто удается обнаружить семантические ошибки модели, проявляющиеся в ненадлежащем поведении сети.

3. ТРАНСЛЯТОР ИЗ SDL В СЕТЕВЫЕ МОДЕЛИ

Алгоритм перевода SDL-спецификаций в сетевые модели системы SPV реализован методом двухпроходной трансляции. На первом проходе модулем синтаксического анализатора строится внутреннее представление спецификации, а на втором — по этому представлению генерируется иерархическая сетевая модель. Синтаксический анализатор языка SDL построен с помощью программного средства Bison.

Транслятор функционирует следующим образом. Модуль анализатора обрабатывает текстовый файл, содержащий SDL-спецификацию, т. е. осуществляет лексическую свертку и синтаксический разбор и строит внутреннее представление спецификации. В случае отсутствия ошибок запускается модуль генерации сетевой модели. Сначала строится внутреннее представление ИВТ-сети, а затем осуществляется ее визуализация в системе SPV. Связь между модулями является однонаправленной и реализуется посредством сервисных функций (без использования глобальных переменных).

Иерархическая сетевая модель (ИВТ-сеть) — это композиция множества неиерархических сетей, называемых страницами. Страницы могут содержать вершины специального типа, которые называются модулями и соединяются с местами на странице по тому же принципу, что и переходы. Модуль представляет подсеть, располагающуюся на отдельной странице, которая в свою очередь может содержать модули.

Такая страница называется подстраницей страницы, на которой располагается модуль. Подстраница содержит копии всех мест, с которыми связан модуль. Место-копия может быть входным местом для некоторого пе-

рехода или модуля на подстранице тогда и только тогда, когда его прототип является входным местом для модуля, представляющего подстраницу. Аналогично, только копия выходного места-прототипа может быть выходным местом некоторого перехода или модуля на подстранице.

Поведение иерархической сети определяется поведенчески эквивалентной ей неиерархической сетью, получающейся при замещении всех модулей страницами, которые они представляют. При этом каждый модуль вместе со своими дугами удаляется со страницы, а на его место помещается подсеть, располагавшаяся на подстранице. Соединение сетей происходит по местам: каждое место-прототип склеивается со всеми своими копиями.

Построение внутреннего представления сети производится по шагам, соответствующим этапам в описании алгоритма трансляции [4]. В первую очередь создается корневой уровень иерархической сети, который состоит из модулей системы SPV, представляющих блоки SDL-спецификации. Также на этом этапе создаются места-очереди, моделирующие каналы. Эти модули и места соединяются дугами в соответствии с описанием SDL-системы. Следующие четыре шага генерации выполняются последовательно для каждого блока.

На втором шаге генерируется сеть, реализующая блок SDL-спецификации. Эта сеть, в свою очередь, содержит модули, соответствующие процессам, и места-очереди, моделирующие маршруты сигналов. Эти модули и места соединяются дугами в соответствии с описанием блока.

На следующем шаге генерируется сеть, реализующая процесс SDL-спецификации. Она содержит модули, соответствующие SDL-переходам процесса, места, моделирующие переменные и таймеры, и некоторые служебные места. На этом этапе построения сети дуги не создаются, а достраиваются на следующем шаге, поскольку нельзя заранее сказать, в каких переходах используется переменная или таймер.

На следующем шаге — трансляции SDL-переходов — создается подсеть, в которой реализуется логика SDL-перехода. В процессе построения этой сети создаются дуги для сети третьего уровня.

На завершающем шаге создаются модули, реализующие процедуры и действия с таймерами, если таковые есть.

При построении сети в системе SPV используются средства создания иерархических сетевых моделей, предоставляемые этой системой. После того как сеть создана, осуществляется размещение ее элементов на плоскости, т. е. каждому элементу приписываются координаты на соответствующих страницах системы SPV. При этом фрагменты сети, реализующие действия SDL-переходов, размещаются типовым образом.

4. ЭКСПЕРИМЕНТ С ПРОТОКОЛОМ INRES

Описание протокола InRes приведено в работе [5] в составе системы InRes на языке SDL, а также в работе [6].

Полученная при трансляции сеть имеет пять уровней вложенности. Сеть первого уровня содержит модули, моделирующие блоки SDL-спецификации. Сети второго уровня вложенности реализуют процессы, а третьего — SDL-переходы. Сети четвертого уровня представляют действия SDL-переходов. Некоторые из этих сетей также содержат модули — соответствующие им подсети пятого уровня моделируют действия с таймерами. Общее число переходов сети, моделирующей спецификацию InRes, равно 266, а мест — 204.

В ходе экспериментов по симуляции этого протокола в системе SPV была подтверждена ошибка, описанная в работе [5].

5. ЗАКЛЮЧЕНИЕ

Автоматическая генерация сетевых моделей коммуникационных протоколов существенно сокращает трудоемкость проведения экспериментов по их верификации, а использование принципа иерархии — поуровневого создания сети — делает возможным построение сетевых моделей для систем реальной сложности. Моделирование протоколов посредством сетей Петри позволяет распознавать семантические ошибки, которые трудно обнаружить стандартными методами тестирования.

При помощи описанного транслятора были получены сетевые модели для кольцевого (Ring) и InRes протоколов, с полученными сетевыми моделями были проведены эксперименты. В ходе экспериментов были установлены семантические ошибки в этих протоколах.

СПИСОК ЛИТЕРАТУРЫ

1. **Карабегов А. В., Тер-Микаэлян Т. М.** Введение в язык SDL. — М.: Радио и связь, 1993. — 184 с.
2. **Specification and Description Language (SDL).** Recommendation Z.100. — ITU-T, 1993.
3. **Алексеев Г. И., Быстров А. В., Куртов С. А., Мыльников С. П., Непомнящий В. А., Окунишникова Е. В., Чубарев П. А., Чурина Т. Г.** Использование сетей Петри для верификации распределенных систем, представленных на языке

- Estelle // Известия РАН. Теория и системы управления. — 1999. — № 5. — С. 105–116.
4. **Churina T. G.** Colored Petri net approach to modeling of SDL specifications // Joint Bulletin of NCC & IIS. Ser.: Comput. Sci. — 2000. — N 13. — P. 18–39.
 5. **Fisher J., Dimitrov E.** Verification of SDL'92 specifications using extended Petri nets // Proc. IFIP 15th Intern. Symp. on Protocol Specification, Testing and Verification. — Warsaw, 1995. — P. 455–458.
 6. **Непомнящий В. А., Алексеев Г. И., Быстров А. В., Куртов С. А., Мыльников С. П., Окунишникова Е. В., Чубарев П. А., Чурина Т. Г.** Верификация Estelle-спецификаций распределенных систем посредством раскрашенных сетей Петри. — Новосибирск, 1998. — С. 103–112.

**Винокуров А.Н., Ильин И.В., Лобив И.В.,
Мурзин Ф.А., Половинко О.Н., Семич Д.Ф.**

ПРОГРАММНЫЙ КОМПЛЕКС, ПРЕДНАЗНАЧЕННЫЙ ДЛЯ ОБРАБОТКИ РЕЗУЛЬТАТОВ, ПОЛУЧЕННЫХ МЕТОДОМ ЯДЕРНОГО КАРОТАЖА НЕФТЯНЫХ СКВАЖИН

ВВЕДЕНИЕ В ЯДЕРНО-МАГНИТНЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ НЕФТЯНЫХ СКВАЖИН

Существующие геофизические методы и технологии оценки текущей и остаточной нефтеводонасыщенности разрабатываемых нефтяных пластов либо недостаточно достоверные, либо дорогостоящие и неоперативные. Это особенно характерно для месторождений нефти с низкой и ультранизкой минерализацией пластовых вод, а также для эксплуатируемых месторождений нефти с закачкой в пласты пресной воды для поддержания пластового давления (некоторые месторождения Башкортостана, Татарстана, Пермской области и др.).

Мировой опыт, прежде всего фирм Шлюмберже, Халибартон и Дрессер Атлас, свидетельствует, что в таких условиях удовлетворительные результаты по оценке текущей и остаточной нефтенасыщенности могут быть получены по данным углеродно-кислородного каротажа, реализуемого с применением современной скважинной программно-управляемой спектрометрической аппаратуры с высокочастотным генератором нейтронов на 14 МэВ, быстрого канала регистрации и телеметрии. А также с помощью энергетического и временного анализа вторичных излучений различной физической природы, создания эффективного программно-методического и метрологического обеспечения [1–6].

Благодаря конверсии ВПК и доступности современной вычислительной техники, подобная аппаратура была разработана ОАО НПП ВНИИГИС в 1997 году, а в 1997–1999 годах проведены научно-исследовательские работы по её промышленной апробации и обработки методики измерений в нефтегазовых скважинах нефтяных месторождений Поволжья.

В настоящем проекте методических рекомендаций обобщен первый опыт применения отечественной технологии для оценки нефтеводонасыщенности коллекторов нефти. В методических рекомендациях изложены физические основы метода, обоснованы аналитические параметры для

оценки коэффициентов нефтеводонасыщенности в различных физико-геологических условиях, рассмотрено влияние мешающих факторов, способы качественной и количественной интерпретации данных С/О-каротажа, аппаратура и программное обеспечение метода.

ФИЗИЧЕСКИЕ ОСНОВЫ УГЛЕРОДНО-КИСЛОРОДНОГО КАРОТАЖА

Физической основой углеродно-кислородного каротажа является процесс неупругого рассеяния быстрых нейтронов на ядрах элементов, приводящих к переходу в активное состояние различных энергетических уровней. Для относительно лёгких породообразующих элементов (от углерода до железа) число возбужденных уровней оказывается небольшим, в силу чего они испускают ограниченное количество гамма-линий. По энергии и интенсивности потока последних представляется возможным идентифицировать и количественно оценить содержание в среде i -го элемента.

Для неупругого рассеяния нейтрона в веществе необходимо, чтобы его энергия была достаточна для прохождения хотя бы самых низких энергетических уровней атомов вещества, т. е. была как минимум $E_n = 1 \div 2.0$ МэВ. Далее замедление нейтронов осуществляется за счёт упругих соударений с ядрами, переходящее затем в термализацию и диффузию надтепловых и тепловых нейтронов. Таким образом, все перечисленные физические процессы, непрерывно сменяя друг друга, разделены во времени. Из теоретических оценок следует, что в горных породах длительность процесса неупругого рассеяния составляет порядка $1.0 \div 4.0$ мкс, упругого рассеяния и термализации до $50 \div 100$ мкс и диффузии до $1000 \div 2000$ мкс.

Основные породообразующие элементы могут быть идентифицированы по спектрам ГИНР и ГИРЗ на основе оптимальной временной и энергетической селекции гамма-излучения.

Наиболее эффективным параметром для оценки нефтенасыщенности пластов может являться величина отношения потоков гамма-излучения от ядер углерода и кислорода в спектрах ГИНР, закономерно изменяющаяся как в терригенных, так и карбонатных коллекторах в зависимости от их пористости и элементного состава.

Приведённые на рис. 1 зависимости представляют собой отношение массовых долей элементов в моделях силикатного состава, имитирующие нефтеводонасыщенные пласты. В спектрах ГИРЗ наиболее контрастно проявляются такие элементы, как водород, хлор, железо, кремний, кальций, с меньшей вероятностью — сера, магний.

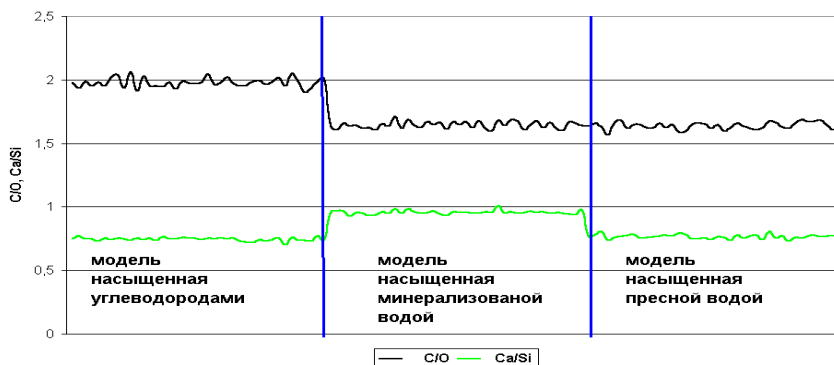


Рис. 1. Показание параметров C/O и Ca/Si в моделях силикатного состава различной насыщенности при $K_n=40\%$

Поэтому эта информация представляет интерес, в первую очередь, для выделения и оценки пористости пластов и определения компонентного состава горных пород, слагающих разрезы скважин.

МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ РАСЧЁТА ИНДЕКСОВ ДЛЯ ОПРЕДЕЛЕНИЯ НЕФТЕСОДЕРЖАНИЯ ПЛАСТА

Энергетическая привязка спектров

На первом этапе обработки энергетических спектров необходимо произвести энергетическую привязку. По характерным фотопикам произвести интерполяцию, при которой ось абсцисс каналов (амплитуда зарегистрированных сигналов в единицах АЦП) отображается в ось энергий путем линейной интерполяции. Все последующие расчеты выполняются исходя из пары ось абсцисс–ось энергий и не требует вмешательства оператора.

Сложность данной процедуры состоит в том, что данный процесс имеет большой субъективный фактор. Интерпретатор должен хорошо знать взаимодействие нейтронов с веществом как быстрых нейтронов, так и тепловых. Он должен уметь различать характерные фотопики на спектрах неупругого рассеяния, радиационного захвата и наведенной активности. Он также должен знать энергию гамма-квантов естественной радиоактивности (ряд урана, ряд тория, калий), правильно их интерпретировать.

Все это накладывает большие требования к знаниям в ядерной физике и геофизике и опыту при обработке энергетических спектров. При этом правильность выполнения энергетической привязки может иметь погрешность в силу субъективных и технических причин. К примеру, изменение спектра при низкой температуре неэффективно из-за низкой статистики высокоэнергетических гамма-квантов.

Алгоритм, который будет описан ниже, имеет целью повысить точность энергетической привязки не менее чем в два раза и устранить субъективный фактор при решении данной задачи.

Обрабатываемые данные

Входными данными являются энергетические спектры различных видов. Типичный энергетический спектр приведен ниже.

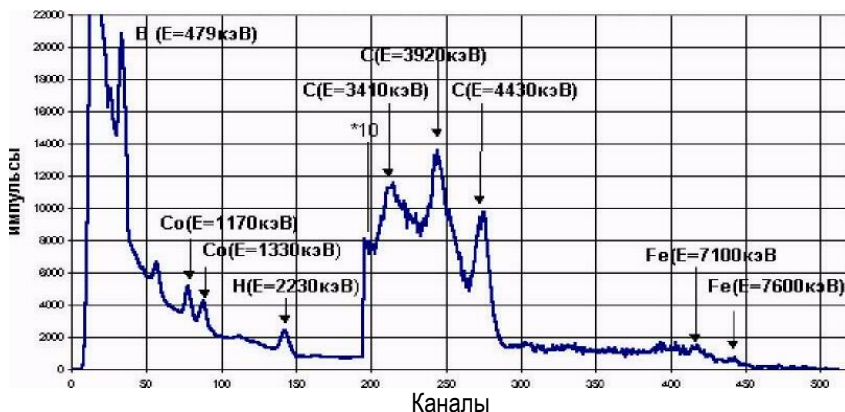


Рис. 2. Спектр гамма излучения полученный от источников CO-GO, Po-Be

По оси абсцисс расположены каналы, или энергии вспышек. По оси ординат откладывается количество зарегистрированных вспышек на данном интервале времени. Для большей наглядности, начиная со 180-ого канала, амплитуда увеличена в 10 раз.

На рис. 2 видны фотопики, соответствующие отдельным элементам. Некоторым элементам соответствует несколько фотопиков. Например, железу соответствует три фотопика. Два из них показано на графике. Третий, находящийся левее данных двух, не показан. Он почти не проявляется в силу наличия шумов.

Обработка временных спектров

Типичный временной спектр приведён на рис. 3. Известно, что с точностью до шумов небольшой амплитуды спектр состоит из трёх частей, характеризующих отклик от ближней, средней и дальней зон.

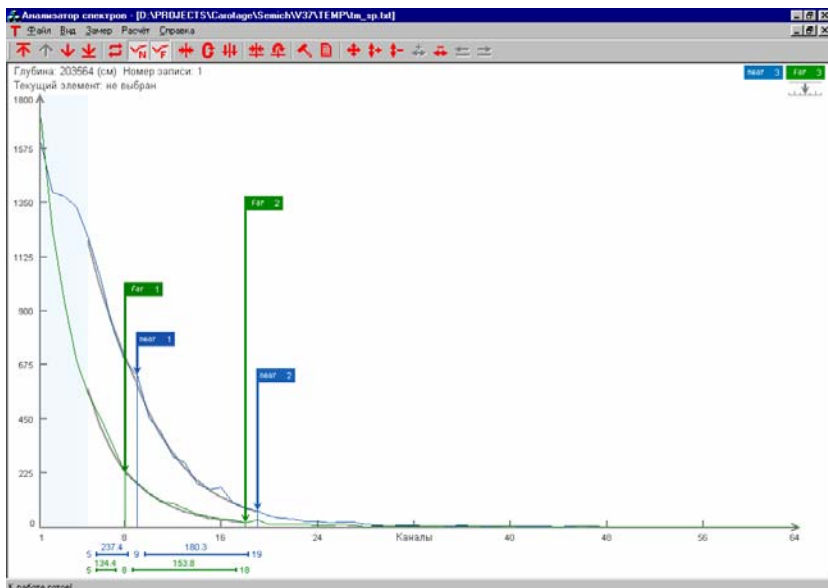


Рис. 3. Типичный временной спектр

Обозначим временной спектр $T(i)$, где i — номер отсчёта по времени, в нашем случае $1 \leq i \leq 64$. В типичных случаях существуют точки $i_0 < i_1 < i_2 < i_3$ такие, что на каждом из интервалов $I_k = [i_{k-1}, i_k]$, $k = 1, 2, 3$ спектр представляет собой экспоненту вида $f_k(x) = e^{-\lambda_k(x+x_0^k)}$, $k = 1, 2, 3$.

Далее можно считать, что $i_0 = 4$. При $i = 1, 2, 3$ мы имеем неустойчивость работы аппаратуры.

Известно также, что в типичных случаях i_1, i_2, i_3 находятся в окрестностях точек 10, 20, 30 соответственно. Значения $T(i)$ при $i > 35$ не представляют интереса.

Для того чтобы локализовать i_1 , i_2 , используется дискретное вейвлет-преобразование, построенное на основе вейвлета Добеши DB4.

На спектре $S(i)$ вейвлет-преобразования в окрестностях точек 10 и 20 имеются пики большой амплитуды. Они локализуют место изменения параметра λ , т.е. i_1 , i_2 .

Более того, полагаем i_1 равное такому i , что $8 \leq i \leq 12$ и $|S(i)|$ достигает максимума в указанной окрестности. Заметим, что во всех примерах, которые были просчитаны, такая точка была только одна. Точку i_2 выбираем аналогично на интервале $27 \leq i \leq 33$.

Вблизи точки 30 на спектре также имеются два локальных максимума наибольшей амплитуды. Часто они имеют разные знаки.

Детальный анализ показывает, что они не характеризуют место изменения параметра λ . Их появление вызвано тем, что более высокочастотные гармоники «загашивают» неточность приближения исходного сигнала линейной комбинацией низкочастотных гармоник. Поэтому для нахождения i_3 используется другой метод.

Сначала подсчитаем суммы $\sum_k = \sum_{j=k}^{64} T(j)$, $10 \leq k \leq 64$.

Легко видеть, что $\sum_{10} \geq \sum_{11} \geq \dots \geq \sum_{64}$. В программной реализации указанные сумм подсчитываются в обратном порядке.

Пусть $M = \sum_{10}$, C — некоторая константа, причём $C = 1/10$. Находим:

$$\sum_{k-1} > C \cdot M \geq \sum_k.$$

Если $k > i_2$, то полагаем $i_3 = k$. В противном случае полагаем $i_2 = k$ и считаем, что спектр распадается не на три, а на два участка $[i_0, i_1]$, $[i_1, i_2]$. Отметим ещё, что k не может быть расположено левее i_1 , так как там значения $T(i)$ большие, а вся описанная выше технология направлена на «отсечение» хвоста, почти сплошь состоящего из нулей.

Дальнейшая обработка состоит в следующем. Переходим к спектру, определённому не на номерах каналов (т.е. на номерах временных отсчётов) к спектру, а определённому на реальном времени в микросекундах, полагая

$$\mathbf{T}(x_i) = T(i), \quad x_i = 60 + i \cdot 40.$$

Имеется три соответствующих множества

$$X_1 = \{x_i : i_0 \leq i \leq i_1\},$$

$$X_2 = \{x_i : i_1 \leq i \leq i_2\},$$

$$X_3 = \{x_i : i_2 \leq i \leq i_3\}.$$

На каждом из участков

$$\mathbf{T}(x_i) \approx e^{\lambda_k(x+x_0^s)}, \quad k=1,2,3.$$

Поэтому параметры экспонентов можно вычислить, используя логарифмирование и метод наименьших квадратов для линейной функции.

Нас интересуют величины $\tau_k = 1/\lambda_k$, $k=1, 2, 3$, характеризующие времена жизни гамма-квантов, приходящих из ближней, средней и дальней зон.

В случае, когда средняя зона вырождается, сливаясь с ближней, получаем два значения τ_1, τ_2 .

В алгоритме предусмотрена возможность усреднения значений λ , соответственно τ аналогично медианной фильтрации.

Фиксируем небольшое h , обычно $h=3, 4, 5$.

Полагаем $i_0=4$, i_1 находим при помощи вейвлет-преобразования. Далее рассмотрим отрезки вида $[i_0, i]$, $i-h \leq i \leq i_1+h$.

Для каждого такого отрезка подсчитываем своё λ , заметим, что их нечётное число. Полагаем λ_1 равным медиане из полученных значений λ . Это означает, что если все данные λ упорядочим по возрастанию, то h штук будет $\leq \lambda_1$ и h штук $\geq \lambda_1$. Именно так работает медианный фильтр, подавляющий импульсные помехи.

Соответственно корректируем i_1 : берём именно то i , которому соответствует данное медианное. На следующем участке действуем аналогично. Мы имеем новое i_1 и рассматриваем отрезки вида

$$[i_2, i], \quad i_2-h \leq i \leq i_2+h.$$

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ РАСЧЁТА ИНДЕКСОВ И ДРУГИХ ФИЗИЧЕСКИХ ВЕЛИЧИН ДЛЯ ОПРЕДЕЛЕНИЯ НЕФТЕСОДЕРЖАНИЯ ПЛАСТА

Программа «Анализатор спектров» запоминает форму фотопиков, которые пользователь указывает вручную и далее, используя их форму, осуществляет начальную привязку на всех замерах по глубине. В правой части отображается изменение положения фотопика, которое определила программа. Основное окно программы представлено на рис. 4.

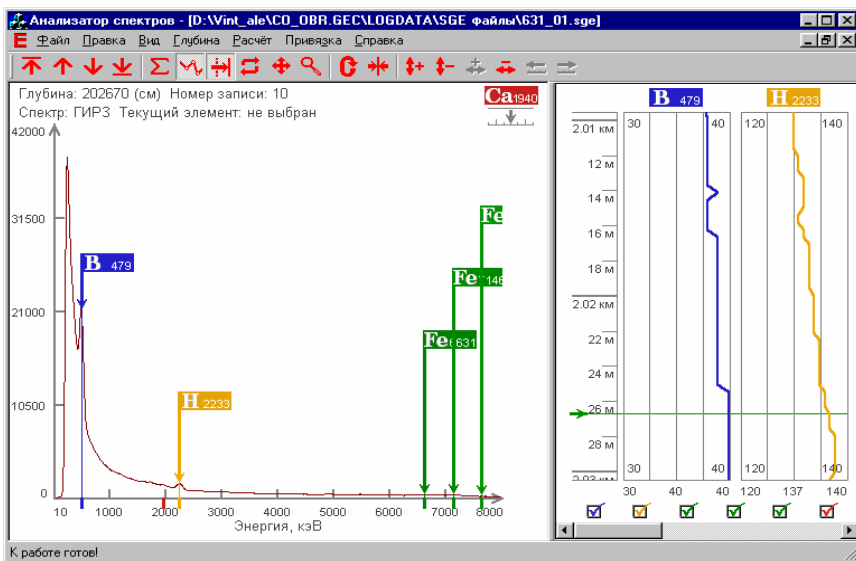


Рис. 4. Основное окно программы «Анализатор спектров»

Отметим, что на горизонтальной оси теперь отображаются энергии в кэВ, а не каналы. Также на оси энергии вертикальными штрихами (маркерами) отмечено, где должны располагаться определяемые для энергетической привязки прочие (неосновные) фотопики. Маркеры и одноименные флажки имеют одинаковый цвет.

Программа позволяет проконтролировать качество привязки, сделанной программой, и внести свои поправки. Для этого необходимо пролистать спектры, двигаясь по замерам вниз по глубине, и проконтролировать положение курсоров на фотопиках и изменение показания кривой динамики положения фотопика в правой части.

После приведения спектров к энергетической шкале и учёта теплового дрейфа переходят непосредственно к расчету основных показателей.

Для дальнейшего использования расчетные данные могут быть сохранены в LAS-формате. Этот формат поддерживают большинство геофизических интерпретационных пакетов программ. Для этого существует в меню пункт Файл, «Экспорт», причем подменю «Экспорт» позволяет сохранить расчетные данные в LAS-формате «Экспорт результатов...», а также сохранить приведенные и энергетически привязанные спектры ГИНР и ГИРЗ. «Экспорт ГИНР и ГИРЗ...» осуществляет экспорт спектров в текстовый формат.

СПИСОК ЛИТЕРАТУРЫ

1. **Новиков Г.Ф., Копков Ю.Н.** Радиоактивные методы разведки. — Л.: «Недра», 1965. — 755 с.
2. **Хуснуллин М.Х.** Геофизические методы контроля разработки нефтяных пластов. — М.: «Недра», 1989. — 188 с.
3. **Резванов Р.А.** Радиоактивные и другие неэлектрические методы исследования скважин. — М.: «Недра», 1982. — 367 с.
4. **Алексеев Ф.А., Головацкая И.В., Гулин Ю.А. и др.** Ядерная геофизика при исследовании нефтяных месторождений. — М.: «Недра», 1978. — 359 с.
5. **Фертл В.Х.** Спектрометрия естественного гамма-излучения в скважине // Нефть, газ и нефтехимия за рубежом. — 1986. — № 3–11.
6. **Kozhevnikov D.A., Shagin V.L.** A method of treating the spectral response of a tool in open and cased boreholes to determine the natural radioactivity of rocks // Nucl. Geophys. — 1989. — Vol.3, №. 1. — P. 17–29.

Грибовская Наталия Сергеевна

ТЕОРЕТИКО-КАТЕГОРНАЯ ХАРАКТЕРИЗАЦИЯ ЯЗЫКОВЫХ ЭКВИВАЛЕНТНОСТЕЙ ВРЕМЕННЫХ ПАРАЛЛЕЛЬНЫХ МОДЕЛЕЙ

1. ВВЕДЕНИЕ

В последнее десятилетие методы теории категорий активно используются для описания и изучения параллельных систем и процессов. Категория включает в себя множество моделей и морфизмы между ними, которые представляют собой некоторый вид моделирования. Часто такая теория используется для сравнения различных моделей.

Одним из наиболее важных понятий теории параллельного программирования является понятие эквивалентности между процессами. Поведенческие эквивалентности обычно используются при спецификации и верификации с целью сравнения поведения систем, а также упрощения их структуры. Наиболее известными из них являются языковые эквивалентности [1]. В этом случае эквивалентность формулируется в терминах равенства языков систем.

В работе [3] был предложен новый теоретико-категорный подход к исследованию эквивалентностей на примере бисимуляции. В рамках этого подхода эквивалентность представляется конструкцией открытых морфизмов. В дальнейшем этот подход стал использоваться и для определения других эквивалентностей (языковой, слабой и сильной бисимуляционной, бисимуляционной с сохранением истории и т.д.) [6].

В настоящее время резко возрос интерес к разработке и исследованию распределенных систем, функционирующих в режиме реального времени. Поэтому в литературе были сделаны попытки ввести понятие времени в эквивалентные отношения, чтобы позволить исследовать временные аспекты поведения систем. Так, например, в статье [2] методами теории категорий была решена проблема разрешимости временной бисимуляции для временных интерливинговых моделей — временных систем переходов, в работе [11] была приведена теоретико-категорная ха-

* Данная работа выполнена при поддержке Министерства образования Российской Федерации, грант № А03-2.8-353.

рактизация временной интерливинговой языковой эквивалентности, а в статье [4] — временной частично-упорядоченной эквивалентности Пратта.

Цель данной работы состоит в разработке основы для построения временной частично упорядоченной языковой эквивалентности в контексте моделей реального времени с семантикой “истинного параллелизма” — временных структур событий (семантика “истинного параллелизма” в отличие от интерливинговой позволяет явным образом описывать и изучать отношение параллелизма между событиями систем). Исследуемая эквивалентность формулируется в терминах равенства временных частично-упорядоченных языков системы. Такая эквивалентность сильнее частично-упорядоченной эквивалентности Пратта в том смысле, что любые языково-эквивалентные временные структуры событий также будут являться и частично-упорядоченно эквивалентными по Пратту.

Статья организована следующим образом. Во втором разделе описывается модель временных структур событий и определяется временная языковая эквивалентность. В разд. 3 строится категория временных структур событий и приводятся некоторые свойства построенной категории. Разд. 4 посвящен теоретико-категорной характеристике. В частности, вводится понятие открытого морфизма и приводится его критерий. Далее определяется абстрактная эквивалентность в терминах существования симметричной конструкции открытых морфизмов и показывается совпадение временной языковой эквивалентности с введенной абстрактной эквивалентностью. Заключение можно найти в разд. 5.

2. ВРЕМЕННЫЕ СТРУКТУРЫ СОБЫТИЙ

В этом разделе определяются основные понятия и определения, касающиеся временных структур событий и языковой эквивалентности между ними.

Сначала напомним определение структуры событий [9]. Структура событий состоит из множества событий вместе с отношением причинной зависимости между ними. Более того, в модель введено отношение конфликта между событиями, что делает модель недетерминированной. Функция меток фиксирует действие, которое соответствует событию.

Пусть L — конечное множество действий. (Помеченной) структурой событий над L называется четверка $S = (E, \leq, \#, l)$, где E — это счетное множество событий; $\leq \subseteq E \times E$ — это частичный порядок (от-

ношение причинной зависимости), удовлетворяющий принципу конечности причин: $\forall e \in E \circ \{e' \in E \mid e' \leq e\}$ — конечное множество; $\# \subseteq E \times E$ — симметричное и иррефлексивное отношение (отношение конфликта), удовлетворяющее принципу наследования конфликта: $\forall e, e', e'' \in E \circ e \# e' \leq e'' \Rightarrow e \# e''$; $l: E \rightarrow L$ — функция меток.

Для структуры событий $S = (E, \leq, \#, l)$ определим $\sim = (E \times E) \setminus (\leq \cup \leq^{-1} \cup \#)$ — отношение параллелизма. Пусть $C \subseteq E$. Тогда C — лево-замкнутое тогда и только тогда, когда $\forall e, e' \in E \circ e \in C \wedge e' \leq e \Rightarrow e' \in C$; C — бесконфликтное тогда и только тогда, когда $\forall e, e' \in C \circ \neg(e \# e')$; C — конфигурация S тогда и только тогда, когда C лево-замкнуто и бесконфликтно. Пусть $\mathcal{C}(S)$ обозначает множество всех конечных конфигураций S . Для $C \subseteq E$ ограничение S на C (обозначается $S[C]: (C, \leq \cap (C \times C), \# \cap (C \times C), l|_C)$). Далее будем использовать запись \emptyset для обозначения пустой структуры событий $(\emptyset, \emptyset, \emptyset, \emptyset)$. Структура событий S называется структурой событий без автопараллелизма, если $\forall e, e' \in E_S \circ ((e \sim_S e' \wedge l_S(e) = l_S(e')) \Rightarrow e = e')$. Далее будут рассматриваться только структуры событий без автопараллелизма.

Теперь представим модель временных структур событий, которая расширяет модель структур событий путем добавления раннего и позднего (относительно глобального времени) моментов выполнения каждого события. Выполнение временной структуры событий называется *временной конфигурацией*. Зафиксируем два множества, с которыми будем работать в дальнейшем. Пусть \mathbf{N} обозначает множество натуральных чисел, а \mathbf{R} — множество неотрицательных вещественных чисел.

Определение 1. (Помеченная) временная структура событий над L — это тройка $TS = (S, Eot, Lot)$, где $S = (E, \leq, \#, l)$ — (помеченная) структура событий над L ; $Eot, Lot: E \rightarrow \mathbf{R}$ — функции раннего и позднего момента выполнения событий, удовлетворяющие $Eot(e) \leq Lot(e)$ для всех $e \in E$.

Временная структура событий имеет корректное таймирование, если $e' \leq_S e \Rightarrow Eot(e') \leq Eot(e)$, и $Lot(e') \leq Lot(e)$ для всех $e, e' \in E$. Далее рассматриваются только временные структуры событий с корректным таймированием.

Для изображения временных структур событий используются следующие соглашения. Метки-действия и временные ограничения, связанные с событием, рисуются рядом с последним. Если это не мешает пониманию, то для идентификации события будут использоваться только метки-действия.

Отношение причинной зависимости графически изображается дугами (исключая те, что получены по транзитивности), кроме того, рисуется отношение конфликта (исключая те, что получены по принципу наследования конфликта). Следуя этим соглашениям, приведем тривиальный пример временной структуры событий (см. рис. 1).

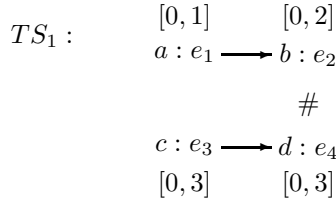


Рис. 1. Пример временной структуры событий

Временные структуры событий TS и TS' *изоморфны* ($TS \simeq TS'$), если существует биекция $\varphi : E_{TS} \longrightarrow E_{TS'}$ такая, что $e \leq_{TS} e'$ тогда и только тогда, когда $\varphi(e) \leq_{TS'} \varphi(e')$, $e \#_{TS} e'$ тогда и только тогда, когда $\varphi(e) \#_{TS'} \varphi(e')$, $l_{TS}(e) = l_{TS'}(\varphi(e))$, $Eot_{TS}(e) = Eot_{TS'}(\varphi(e))$ и $Lot_{TS}(e) = Lot_{TS'}(\varphi(e))$ для всех $e, e' \in E_{TS}$.

В данной работе рассматривается частично-упорядоченная семантика, основанная на временных *rom*-множествах. Сначала определим *временное частично-упорядоченное множество* как временную структуру событий $TP = (S_{TP} = (E_{TP}, \leq_{TP}, \#_{TP}, l_{TP}), Eot_{TP}, Lot_{TP})$ такую, что $\#_{TP} = \emptyset$ и $Eot_{TP}(e) = Lot_{TP}(e)$ для всех $e \in E_{TP}$. Изоморфные классы временных частично-упорядоченных множеств называются *временными rom-множествами*. Пустым *временным rom-множеством* является $(\mathcal{O}, \emptyset, \emptyset)$. Будем использовать запись \mathcal{TProm}_L для обозначения множества конечных временных *rom*-множеств, помеченных над L .

Пусть $TS = (S, Eot, Lot)$ — временная структура событий, $C \in \mathcal{C}(S)$, и $T : C \longrightarrow \mathbf{R}$. Тогда $TC = (C, T)$ есть *временная конфигурация* TS тогда и только тогда, когда выполнены следующие условия:

- (i) $\forall e \in C \diamond Eot(e) \leq T(e) \leq Lot(e)$;
- (ii) $\forall e, e' \in C \diamond e \leq_{TS} e' \Rightarrow T(e) \leq T(e')$.

Неформально, временная конфигурация состоит из конфигурации и временной функции, записывающей моменты выполнения событий относительно глобального времени. Эта функция удовлетворяет некоторым дополнительным требованиям: (i) событие может выполняться только в те моменты времени, которые заданы ее временными ограничениями; (ii) для всех событий e и e' , которые произошли, если e' причин-

но зависит от e , то e должна произойти раньше e' . *Начальная временная конфигурация* TS — это $TC_{TS} = (\emptyset, \emptyset)$. Будем писать $TC(TS)$ для обозначения множества всех временных конфигураций TS . Чтобы проиллюстрировать это понятие, рассмотрим множество всех возможных временных конфигураций для временной структуры событий TS_1 , изображенной на рис. 1: $\{(\emptyset, \emptyset), (\{e_1\}, T_1), (\{e_3\}, T_2), (\{e_1, e_3\}, T_3), (\{e_1, e_2\}, T_4), (\{e_3, e_4\}, T_5), (\{e_1, e_2, e_3\}, T_6), (\{e_1, e_3, e_4\}, T_7), | T_1(e_1) \in [0, 1]; T_2(e_3) \in [0, 3]; T_3(e_1) \in [0, 1], T_3(e_3) \in [0, 3]; T_4(e_1) \in [0, 1], T_4(e_2) \in [0, 2], T_4(e_1) \leq T_4(e_2); T_5(e_3), T_5(e_4) \in [0, 3], T_5(e_3) \leq T_5(e_4); T_6(e_1) \in [0, 1], T_6(e_2) \in [0, 2], T_6(e_3) \in [0, 3], T_6(e_1) \leq T_6(e_2); T_7(e_1) \in [0, 1], T_7(e_3), T_7(e_4) \in [0, 3], T_7(e_3) \leq T_7(e_4)\}$.

Для временной конфигурации $TC = (C, T)$ *ограничение* TS на TC (обозначается как $TS \upharpoonright TC$) определено как изоморфный класс $(S \upharpoonright C, T)$. Для $TC_1 = (C_1, T_1), TC_2 = (C_2, T_2) \in TC(TS)$ будем писать $TC_1 \longrightarrow TC_2$, если и только если $C_1 \subseteq C_2$, $T_2|_{C_1} = T_1$, и $\forall e \in C_1 \forall e' \in (C_2 \setminus C_1) \diamond T_1(e) \leq T_2(e')$; $TC_1 \xrightarrow{TP} TC_2$ тогда и только тогда, когда $TC_1 \longrightarrow TC_2$ и TP — изоморфный класс $(S_{TS} \upharpoonright (C_2 \setminus C_1), T_2|_{(C_2 \setminus C_1)})$.

Множество $L_{tp}(TS) = \{TP \mid TC_{TS} \xrightarrow{TP} TC \text{ для некоторого } TC \in TC(TS)\}$ называется *временным частично-упорядоченным языком* TS (*тр-язык*).

Теперь приведем определение языковой эквивалентности, основанное на временных *рот*-множествах (TP -эквивалентность), на множестве временных структур событий.

Определение 2. *Временные структуры событий* TS и TS' TP -эквивалентны (обозначается $TS \equiv_{TP} TS'$) тогда и только тогда, когда $L_{tp}(TS) = L_{tp}(TS')$.

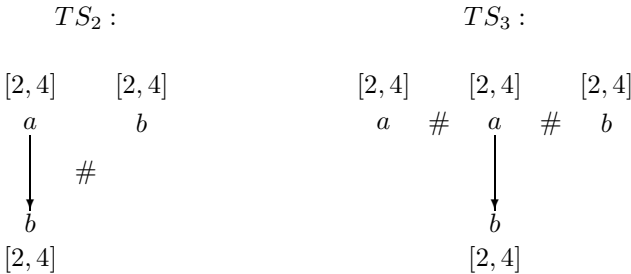


Рис. 2. TP -эквивалентные временные структуры событий

Рассмотрим временные структуры событий, изображенные на рис. 2, имеем $TS_2 \equiv_{TP} TS_3$, но $TS_1 \not\equiv_{TP} TS_2$, так как временное *rot*-множество $\overset{[t_1, t_1]}{c}$ ($t_1 \in [0, 3]$) принадлежит $L_{tp}(TS_1)$, но не принадлежит $L_{tp}(TS_2)$.

3. КАТЕГОРИЯ ВРЕМЕННЫХ СТРУКТУР СОБЫТИЙ

В этом разделе определяется категория временных структур событий.

Для начала введем понятие морфизма между временными структурами событий.

Определение 3. Морфизм *между временными структурами событий* $TS = (E, \leq, \#, l, Eot, Lot)$ и $TS' = (E', \leq', \#', l', Eot', Lot')$, $\mu : TS \rightarrow TS'$ — это функция $\mu : E \rightarrow E'$ такая, что:

- $l' \circ \mu = l$;
- $TC = (C, T) \in \mathcal{TC}(TS) \Rightarrow (\mu C, T') \in \mathcal{TC}(TS')$ с $T' \circ \mu = T$, и
 - $\forall e, e' \in C \circ \mu(e) = \mu(e') \Rightarrow e = e'$;
 - $\forall e, e' \in C \circ \mu(e) < \mu(e') \Leftrightarrow e < e'$.

Для временной конфигурации $TC = (C, T)$ будем писать μTC вместо $(\mu C, T')$ с $T' \circ \mu = T$.

Рассмотрим полезное свойство введенного морфизма.

Теорема 1. Пусть $\mu : TS \rightarrow TS'$ — морфизм и TP — временное *rot*-множество. Если $TC_{TS} \xrightarrow{TP} TC$ в TS , тогда $TC_{TS'} \xrightarrow{TP} \mu TC$ в TS' .

Доказательство. Следует из определений морфизма и отношения \xrightarrow{TP} . \square

Следствие 1. Пусть $\mu : TS \rightarrow TS'$ — морфизм и $TS \in \mathcal{TC}(TS)$. Тогда $TS[TC \approx TS'] \mu TC$.

Теперь определим категорию временных структур событий следующим образом.

Определение 4. Временная структура событий (помеченная над L) с морфизмами между ними формирует категорию временных структур событий TS_L , в которой композиция морфизмов есть обычная композиция функций, а тождественный морфизм есть тождественная функция.

Предложение 1. TS_L — категория.

Предложение 2. Для временной структуры событий TS и временного рот-множества $TP = ((E_{TP}, \leq_{TP}, l_{TP}), Eot_{TP})$ любой временной конфигурации TC в TS такой, что $TC_{TS} \xrightarrow{TP} TC$, можно сопоставить только один морфизм $\mu : TP \rightarrow TS$ такой, что

$$\mu (E_{TP}, Eot_{TP}) = TC.$$

4. ХАРАКТЕРИЗАЦИЯ

Следуя статье [3], выберем временные рот-множества и специальные морфизмы между ними в качестве полной подкатегории категории временных структур событий. Пусть \mathcal{TP}_L^* обозначает подкатегорию категории \mathcal{TP}_L , в которой объекты принадлежат \mathcal{TPot}_L , а морфизмы — тождественные морфизмы или морфизмы с пустой областью определения. Теперь воспользуемся общей схемой, предложенной в работе [3], чтобы определить понятие открытого морфизма.

Определение 5. Морфизм $\mu : TS \rightarrow TS'$ в \mathcal{TS}_L называется \mathcal{TP}_L^* -открытым тогда и только тогда, когда из существования морфизмов $\mu' : TP \rightarrow TP'$ в \mathcal{TP}_L^* , $\mu'' : TP \rightarrow TS$ и $\mu''' : TP' \rightarrow TS''$ в \mathcal{TS}_L таких, что $\mu \circ \mu'' = \mu''' \circ \mu'$, следует существование морфизма $\tilde{\mu} : TP' \rightarrow TS$ в \mathcal{TS}_L такого, что $\mu'' = \tilde{\mu} \circ \mu'$ и $\mu''' = \mu \circ \tilde{\mu}$.

Далее приведем характеристику \mathcal{TP}_L^* -открытости для морфизмов.

Теорема 2. Морфизм $\mu : TS \rightarrow TS'$ является \mathcal{TP}_L^* -открытым тогда и только тогда, когда для любого временного рот-множества TP верно: если $TC_{TS'} \xrightarrow{TP} TC'$ в TS' , то $TC_{TS} \xrightarrow{TP} TC$ в TS и $\mu TC = TC'$.

Доказательство. Проводится по стандартной схеме с использованием определения морфизма и Предложения 2. \square

Теперь введем абстрактное понятие бисимуляции, используя открытые морфизмы.

Определение 6. Временные структуры событий TS_1 и TS_2 называются \mathcal{TP}_L^* -бисимуляционными, если и только если существует конструкция \mathcal{TP}_L^* -открытых морфизмов $TS_1 \xleftarrow{\mu} TS \xrightarrow{\mu'} TS_2$ с вершиной TS .

Заметим, что из Теоремы 3 (см. ниже) следует, что \mathcal{TP}_L^* -бисимуляция является отношением эквивалентности.

Теорема 3. Пусть $\mu_1 : TS_1 \rightarrow TS$ и $\mu_2 : TS_2 \rightarrow TS$ — \mathcal{TP}_L^* -открытые морфизмы. Тогда существует временная структура собы-

тий TS_x , и TP_L^* -открытые морфизмы $\mu'_1 : TS_x \rightarrow TS_1$, $\mu'_2 : TS_x \rightarrow TS_2$ такие, что $\mu_1 \circ \mu'_1 = \mu_2 \circ \mu'_2$.

Доказательство. Без ограничения общности считаем, что $TS_i = (E_i, \leq_i, \#_i, l_i, Eot_i, Lot_i)$ для $i \in \{1, 2\}$.

Для начала построим временную структуру событий

$$TS_x = +(TS_{TC_1 \times TC_2} \mid TC_i = (C_i, T_i) \in \mathcal{TC}(TS_i))$$

для всех $i \in \{1, 2\}$, $\mu_1 TC_1 = \mu_2 TC_2$), где $TS_{TC_1 \times TC_2} = (E_{TC_1 \times TC_2}, \leq_{TC_1 \times TC_2}, \#_{TC_1 \times TC_2}, l_{TC_1 \times TC_2}, Eot_{TC_1 \times TC_2}, Lot_{TC_1 \times TC_2})$ определена так:

- $E_{TC_1 \times TC_2} = \{(e_1, e_2)_{TC_1 \times TC_2} \in C_1 \times C_2 \mid \mu_1(e_1) = \mu_2(e_2)\}$;
- $(e_1, e_2)_{TC_1 \times TC_2} \leq_{TC_i} (e'_1, e'_2)_{TC_1 \times TC_2} \iff e_i \leq_i e'_i$ для всех $i \in \{1, 2\}$;
- $\#_{TC_1 \times TC_2} = \emptyset$;
- $l_{TC_1 \times TC_2}((e_1, e_2)_{TC_1 \times TC_2}) = l_i(e_i)$ для некоторого $i \in \{1, 2\}$;
- $Eot_{TC_1 \times TC_2}((e_1, e_2)_{TC_1 \times TC_2}) = T_i(e_i)$ для некоторого $i \in \{1, 2\}$;
- $Lot_{TC_1 \times TC_2}((e_1, e_2)_{TC_1 \times TC_2}) = T_i(e_i)$ для некоторого $i \in \{1, 2\}$.

Здесь и далее, "+" обозначает алгебраическую операцию, которая означает, что все события одной компоненты находятся в отношении конфликта (#-отношение) со всеми событиями другой компоненты. По построению получаем, что TS_x действительно является временной структурой событий (имеющей корректное таймирование).

Далее, определяем отображения $\mu'_i : TS_x \rightarrow TS_i$ следующим образом: $\mu'_i(e_1, e_2) = e_i$ ($i = 1, 2$). Легко проверить, что эти отображения являются морфизмами. Равенство $\mu_1 \circ \mu'_1 = \mu_2 \circ \mu'_2$ немедленно следует из построения TS_x и определения μ'_i ($i \in \{1, 2\}$).

Наконец, покажем, что μ'_i является TP_L^* -открытым морфизмом ($i \in \{1, 2\}$). Предположим, что $TC_{TS_i} \xrightarrow{TP} TC_i$ в TS_i , тогда по Теореме 1 имеем: $TC_{TS} \xrightarrow{TP} \mu_i TC_i$ в TS . Это влечет следующий факт: $TC_{TS_{3-i}} \xrightarrow{TP} TC_{3-i}$ в TS_{3-i} и $\mu_i TC_i = \mu_{3-i} TC_{3-i}$, согласно Теореме 2. Тогда можно построить $TS_{TC_1 \times TC_2}$ как часть TS_x . Легко видеть, что $TC_x = (E_{TC_1 \times TC_2}, Eot_{TC_1 \times TC_2}) \in \mathcal{TC}(TS_x)$ и $TS_x[TC_x \simeq TS_i[TC_i$. Это означает, что $TC_{TS_x} \xrightarrow{TP} TC_x$ в TS_x . Более того, верно следующее: $\mu'_i TC_x = TC_i$ по определению μ'_i . Таким образом, μ'_i действительно TP_L^* -открытый морфизм ($i \in \{1, 2\}$), по Теореме 2. \square

Установим совпадение абстрактной бисимуляции и временной языковой эквивалентности.

Теорема 4. Пусть TS_1 и TS_2 — временные структуры событий. TS_1 и TS_2 TP_L^* -бисимуляционны тогда и только тогда, когда они TP -эквивалентны.

Доказательство.

(\Rightarrow) Пусть $TS_1 \xleftarrow{\mu_1} TS \xrightarrow{\mu_2} TS_2$ — конструкция TP_L^* -открытых морфизмов. Нам нужно показать, что $L_{tp}(TS_1) = L_{tp}(TS_2)$. Докажем, что $L_{tp}(TS_1) \subseteq L_{tp}(TS_2)$. Доказательство того, что $L_{tp}(TS_2) \subseteq L_{tp}(TS_1)$ симметрично. Возьмем произвольное $TP \in L_{tp}(TS_1)$. Это означает, что $TC_{TS_1} \xrightarrow{TP} TC_1$ в TS_1 . По Теореме 2 получаем, что $TC_{TS} \xrightarrow{TP} TC$ в TS и $\mu_1 TC = TC_1$, так как μ_1 — открытый морфизм. По Теореме 1 следует, что $TC_{TS_2} \xrightarrow{TP} \mu_2 TC$ в TS_2 , т. е. $TP \in L_{tp}(TS_2)$. Таким образом, $L_{tp}(TS_1) = L_{tp}(TS_2)$.

(\Leftarrow) Пусть $L_{tp}(TS_1) = L_{tp}(TS_2)$. Без ограничения общности можно считать, что $TS_i = (E_i, \leq_i, \#_i, l_i, Eot_i, Lot_i)$ для $i \in \{1, 2\}$.

Сначала построим структуру $TS = +(TS_{TC_1 \times TC_2} \mid TC_i = (C_i, T_i) \in \mathcal{TC}(TS_i)$ для всех $i \in \{1, 2\}$, $TS_1[TC_1 \simeq TS_2[TC_2)$, где $TS_{TC_1 \times TC_2} = (E_{TC_1 \times TC_2}, \leq_{TC_1 \times TC_2}, \#_{TC_1 \times TC_2}, l_{TC_1 \times TC_2}, Eot_{TC_1 \times TC_2}, Lot_{TC_1 \times TC_2})$ определена так:

- $E_{TC_1 \times TC_2} = \{(e_1, e_2)_{TC_1 \times TC_2} \in C_1 \times C_2 \mid \phi : TS_i[TC_i \longrightarrow TS_{3-i}[TC_{3-i}, \phi \text{ — изоморфизм и } \phi(e_i) = e_{3-i} \text{ для некоторого } i \in \{1, 2\}\}$;
- $(e_1, e_2)_{TC_1 \times TC_2} \leq_{TC_i} (e'_1, e'_2)_{TC_1 \times TC_2} \iff e_i \leq_i e'_i$ для всех $i \in \{1, 2\}$;
- $\#_{TC_1 \times TC_2} = \emptyset$;
- $l_{TC_1 \times TC_2}((e_1, e_2)_{TC_1 \times TC_2}) = l_i(e_i)$ для некоторого $i \in \{1, 2\}$;
- $Eot_{TC_1 \times TC_2}((e_1, e_2)_{TC_1 \times TC_2}) = T_i(e_i)$ для некоторого $i \in \{1, 2\}$;
- $Lot_{TC_1 \times TC_2}((e_1, e_2)_{TC_1 \times TC_2}) = T_i(e_i)$ для некоторого $i \in \{1, 2\}$.

Далее определим отображения $\mu_i : TS \longrightarrow TS_i$ следующим образом: $\mu_i(e_1, e_2) = e_i$ ($i = 1, 2$). Легко проверить, что μ_1 и μ_2 являются морфизмами.

Наконец, проверим, что μ_i — TP_L^* -открытый морфизм ($i \in \{1, 2\}$). Пусть $TC_{TS_i} \xrightarrow{TP} TC_i$ в TS_i . Тогда $TP \in L_{tp}(TS_1)$. Так как $L_{tp}(TS_1) = L_{tp}(TS_2)$, то $TC_{TS_{3-i}} \xrightarrow{TP} TC_{3-i}$ в $TS_{TS_{3-i}}$ для некоторой временной конфигурации TC_{3-i} . Это означает, что $TS_i[TC_i \simeq TS_{3-i}[TC_{3-i}$. Тогда можно построить $TS_{TC_1 \times TC_2}$ как часть TS . Легко видеть, что $TC = (E_{TC_1 \times TC_2}, Eot_{TC_1 \times TC_2}) \in \mathcal{TC}(TS)$, и $TS[TC \simeq TS_i[TC_i$. Это означает, что $TC_{TS} \xrightarrow{TP} TC$ в TS . Кроме того, имеем $\mu_i TC = TC_i$ в силу опреде-

ления μ_i . Таким образом, μ_i — TP_L^* -открытый морфизм ($i \in \{1, 2\}$) по Теореме 2. \square

5. ЗАКЛЮЧЕНИЕ

В этой работе сделана попытка применить методы теории открытых морфизмов [3] при исследовании языковых эквивалентностей временных моделей с семантикой “истинного параллелизма”. В частности, получена теоретико-категорная характеристика наиболее известной эквивалентности — временной языковой эквивалентности — для временных структур событий. В дальнейшем планируется расширить полученный результат на временные варианты частично-упорядоченной бисимуляционной и тестовой эквивалентностей.

СПИСОК ЛИТЕРАТУРЫ

1. **Hoare C.A.R.** Communicating Sequential Processes. — Prentice-Hall, 1985.
2. **Hune T., Nielsen M.** Timed bisimulation and open maps // Lect. Notes Comput. Sci. — Berlin etc., 1998. — Vol. 1450. — P. 378–387.
3. **Joyal A., Nielsen M., Winskel G.** Bisimulation from open maps // Information and Computation. — 1996. — Vol. 127, N 2. — P. 164–185.
4. **Virbitskaite I. B., Gribovskaja N. S.** Open Maps and Trace Semantics for Timed Partial Order Models // Lect. Notes Comput. Sci. — Berlin etc., 2003. — Vol. 2890. — P. 248–259.
5. **Murphy D.** Time and duration in noninterleaving concurrency // Fundamenta Informaticae. — 1993. — Vol. 19. — P. 403–416.
6. **Nielsen M., Cheng A.** Observing behaviour categorically // Lect. Notes Comput. Sci. — Berlin etc., 1996. — Vol. 1026. — P. 263–278.
7. **Nielsen M., Winskel G.** Petri nets and bisimulation // Theor. Comput. Sci. — 1996. — Vol. 153.
8. **Weise C., Lenzkes D.** Efficient scaling-invariant checking of timed bisimulation // Lect. Notes Comput. Sci. — Berlin etc., 1997. — Vol. 1200. — P. 176–188.
9. **Winskel G.** An introduction to event structures // Lect. Notes Comput. Sci. — Berlin etc., 1988. — Vol. 354. — P. 364–397.
10. **Winskel G., Nielsen M.** Models for concurrency // Handbook of Logic in Comput. Sci. — 1995. — Vol. 4.
11. **Москалева Н.** Теоретико-категорная характеристика трассовой эквивалентности временных параллельных моделей. — Новосибирск, 2002. — с.22. — (Препр. / ИСИ СО РАН; N 99).

А. В. Жданова, П. В. Манкевич

СТАТИСТИЧЕСКИЙ ПОДХОД К СЕМАНТИЧЕСКОМУ СТРУКТУРИРОВАНИЮ ПРЕДМЕТНЫХ ОБЛАСТЕЙ ДЛЯ ЗАПРОСОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

1. ВВЕДЕНИЕ

В настоящее время можно выделить два качественно различных подхода к поиску ответов на вопросы в глобальной сети Интернет и локальных Интранетах. В первом, ориентированном на базы данных, подходе фрагменты Сети рассматриваются как базы данных, а во втором подходе, типичном для многих поисковых систем, данные Интернета рассматриваются как множество неупорядоченных естественно-языковых текстов, плохо структурированных и содержащих избыточную информацию.

Исторически, в решениях задачи удовлетворения естественно-языковых запросов первый подход стал применяться ранее второго. Это обусловлено тем, что базы данных получили широкое распространение еще до повсеместного появления Интернета. В решении задачи удовлетворения запросов на естественном языке в контексте первого подхода были проведены теоретические разработки и построены работающие прототипы информационных систем, в частности основанные на реляционных базах данных [2, 3]. Однако, далеко не всю информацию, имеющуюся в распоряжении пользователей на сегодняшний день, можно наиболее выгодно представить в доступных типах баз данных. Например, тогда как перечень результатов футбольного матча структурно соответствует реляционной базе данных, отображение произвольного естественно-языкового текста в реляционную базу данных нетривиально и неэффективно.

В рамках второго подхода к обработке запросов было продемонстрировано, что упорядочивание множества естественно-языковых документов конкретной предметной области в иерархию способствует повышению эффективности в решении задач классификации сообщений и создания автоответчиков для электронной почты [7, 8]. (Для обобщения понятий следует заметить, что задача автоматического ответа на e-mail сводится к задаче информационного поиска в локальном Интранете, так как для автоматического ответа на сообщение требуется произвести поиск ответа во множестве заданных ответов.) И если подойти к практике, то на сегодняшний день большинство популярных поисковых систем (Google, Yahoo, Yandex...)

создают свои собственные древовидные иерархии веб-ресурсов и используют их в поиске информации. Проблема в том, что даже в системах-рекордсменах по величине проиндексированных ресурсов иерархии составляются вручную. Учитывая прогнозируемый рост и развитие Интернета, существующий неавтоматизированный подход оказывается неприемлемым.

Разработки, ведущиеся в направлении автоматического структурирования предметных областей, включают в себя реализацию алгоритмов, построенных на основе Self-Organizing Maps (SOM – это разновидность нейронных сетей) [4], вероятностных моделей, использующих адаптированный для построения иерархии probabilistic latent semantic analysis [7].

К недостаткам использования SOM для автоматического структурирования предметной области относится отсутствие логического обоснования моделей, использующих SOM. Как следствие, задача определения пороговых значений (например, ограничивающих разрастание SOM в определенном направлении) и выбора параметров становится достаточно сложной. Более того, пороговые значения и параметры будут зависеть от конкретного типа набора кластеризуемых документов. К недостаткам работ по использованию SOM для иерархической кластеризации документов относится также отсутствие оценок корректности получившихся древовидных структур и их пригодности к использованию в информационных системах. Оценки корректности можно получить сравнением автоматически сгенерированной структуры со структурой, составленной экспертом вручную на тех же данных. Оценки пригодности автоматически образованных древовидных структур можно получить после использования получившихся структур в приложениях, выполняющих конкретные задания, качество выполнения которых относительно легко проконтролировать. К таким приложениям, например, относится поисковая система-автоответчик, удовлетворяющая запросы пользователей.

В работе Винокурова и Джиролами [7] оценки корректности, полученные в рамках решения задачи классификации документов, показывают, что использование предложенных вероятностных моделей оправдывает себя в решении данной задачи. Однако вопрос пригодности данных моделей для удовлетворения запросов на естественном языке не рассматривается, у построенной системы нет естественно-языкового интерфейса, алгоритмы для информационного поиска по конкретному запросу не предложены и не опробованы.

В нашей статье мы представляем статистический подход к построению структуры (иерархии) предметной области. Данный подход ориентирован на использование получившейся структуры для поиска информации на ес-

тественном языке. В разд. 2 нашей статьи мы обсуждаем понятие «иерархия» и описываем принципы её построения, в разд. 3 приводим алгоритм построения структуры предметной области, в разд. 4 характеризуем используемый естественно-языковой интерфейс, в разд. 5 приводим результаты тестирования алгоритма и делаем заключение.

2. ИЕРАРХИЯ И ЕЁ ПРЕИМУЩЕСТВА

Во всех относящихся к рассматриваемой проблеме известных нам работах вообще и в нашей работе в частности текстовые документы упорядочиваются в иерархию (конкретнее, в дерево) на основе частичного порядка «категория-подкатегория», введенного на множестве документов, где каждый документ представляет собой категорию, т.е. характеризует некое идентифицируемое понятие, объект или явление. Примерами отношения частичного порядка «категория-подкатегория» являются отношения «is-a» и «part-of». Очевидно, что значительное преимущество в использовании иерархически упорядоченной предметной области в задачах информационного поиска состоит в том, что в самой иерархии заложена информация, которая может выгодно использоваться в поисковых алгоритмах. Например, из иерархии можно узнать, что категория «кошачьи» является подкатегорией категории «животные», и после этого значительное количество времени будет сэкономлено на том, что мы будем искать требуемую нам категорию «львы», сразу направившись по правильной ветке иерархии «животные/кошачьи/...», а не перебирая вместо этого все подкатегории прочих разделов: «птицы», «рептилии» и т.д.

Мы предлагаем статистический алгоритм для автоматического структурирования предметных областей, основанный на понятии *веса текстового документа*. Вес текстового документа в конкретной предметной области определяется количеством и частотностью значимых слов этой области. Чем больше значимых понятий встречается в документе, тем тяжелее для понимания этот документ [9]. Соответственно, легкие для понимания документы описывают общеизвестные концепции с помощью общеупотребительной лексики. Таким образом, после структурирования предметная область представляется деревом категорий, в котором «легкие» документы расположены ближе к корню, а наиболее «тяжелые» находятся в листьях. В свою очередь, решение об установлении отношения «категория-подкатегория» между двумя документами осуществляется, исходя из величины *меры семантической близости* двух документов. Примеры того, как

можно определять вес текстового документа и вычислять величину меры близости двух документов, показаны на рис. 1 и 2.

ВЕС ТЕКСТОВОГО ДОКУМЕНТА

Правило Зипфа: трудность понимания слова $V \sim \frac{1}{Fr(V)}$

Примеры функций $w(X)$, отражающих «легкость» восприятия документа:

$$w(X) = \frac{1}{\prod x_i}, \quad w(X) = \frac{-\sum \ln(x_i)}{n}, \quad w(X) = \sum \frac{x_i x_i^*}{\sqrt{\sum (x_i x_i^*)^2}}$$

Рис. 1. Вес текстового документа X может выражаться через частоты встречающихся в нем слов $\{x_i\}$, общее количество известных слов n , и быть основанным на законе Зипфа: «трудность понимания слова обратно пропорциональна его частоте»

ПРИМЕРЫ МЕР БЛИЗОСТИ ТЕКСТОВЫХ ДОКУМЕНТОВ

Jaccard Association $sim(X, Y) = \left(1 - \frac{a}{a + b + c}\right)^{-1}$,

где $a = \|X \cap Y\|$, $b = \|X \setminus Y\|$, $c = \|Y \setminus X\|$

Taxonomic Distance $sim(X, Y) = \left(\sum_{i=1}^{\|F\|} (x_i - y_i)^2\right)^{-1/2}$

Cosine Measure $sim(X, Y) = \frac{\sum_{1 \leq i \leq \|F\|} x_i y_i}{\sqrt{\sum_{1 \leq i \leq \|F\|} x_i^2 \sum_{1 \leq i \leq \|F\|} y_i^2}}$

Рис. 2. Меры семантической близости текстовых документов могут выражаться через частоты встречающихся в документах слов $\{x_i\}$, $\{y_i\}$, суммирование ведется по мощности множества F слов предметной области

Следовательно, отношение «категория-подкатегория» в нашей древо-видной иерархии можно определить как «от простого к сложному», «от общего к частному», «от известного к неизвестному». Логическое обоснование выбора этого отношения состоит в том, что оно наиболее верно отражает процесс обучения человека. Сначала человек «запрашивает» и узнает «простые» вещи (они быстро находятся на вершине иерархии). Потом, если он не потерял интерес к предметной области, его словарный запас в этой предметной области начинает расширяться, и он «запрашивает» более «сложные» вещи более «сложными» словами. «Сложные» вещи нужно искать в глубине иерархии, т.е. дольше, но и запросы, адресуемые к «сложным» и редким документам из глубин иерархии, встречаются реже. Таким образом, наши принципы конструирования иерархии позволяют естественно-языковому интерфейсу, используемому в системе-автоответчике, эффективно выдавать в качестве ответа документы, соответствующие уровню осведомленности пользователя о предметной области, который, в свою очередь, определяется исходя из «сложности» запроса.

3. АЛГОРИТМ ПОСТРОЕНИЯ СТРУКТУРЫ ПРЕДМЕТНОЙ ОБЛАСТИ

Вход: множество документов.

Выход: иерархически структурированное множество документов.

1. Определить вес каждого документа согласно функции веса.
2. Выбрать самый легкий документ и разместить его в корень иерархии.
3. Пусть в иерархии размещено n документов. Выбрать самый «легкий» документ d из оставшихся неразмещенных. Вычислить значения функции семантической близости между d и $\{d_i\}$ — множеством размещенных документов.
4. Установить отношение «категория-подкатегория» между документом d_{max} , наиболее сходным с документом d , и документом d . Разместить документ d в иерархии как подкатеорию документа d_{max} .
5. Если остались неразмещенные документы, повторить шаг 4, иначе алгоритм завершен.

В данном алгоритме функции веса документа и близости двух документов являются параметрами, т. е. разработчик может их выбрать из известных (в том числе приведенных ранее) функций или построить свои собственные, исходя из своих обстоятельств (см. результаты тестирования в табл. 1).

Таблица 1

**Результаты тестирования алгоритма структуризации
в зависимости от функций веса документа и мер сходности**

Веса	Меры близости							
	CD	CC	JA	SP	SI	SDA	TD	Cosine
entr	61.83	58.2	72.56	77.66	58.3	71.56	75.83	66.16
	11.73	12.41	13.50	14.24	10.52	13.02	14.07	11.97
stat	64.2	57.1	64.13	78.60	59.23	79.8	77.86	66.26
	12.28	15.95	11.84	14.39	12.05	14.58	14.44	13.15
atn	64.6	55.7	65.03	69.83	49.53	76.3	72.96	71.46
	9.79	9.40	12.80	12.39	8.73	12.53	13.72	12.61
bnn	65.23	60.6	74.96	72.16	51.23	75.63	78.13	69.6
	12.07	14.04	14.02	12.91	10.01	13.82	14.11	13.16
ntc	68.2	72.26	65.86	71.23	63.8	76.3	81.16	67.99
	11.82	13.75	11.78	12.73	11.31	13.46	14.80	11.29
npn	65.2	61.0	53.53	68.13	58.16	75.13	68.26	62.7
	17.60	11.90	10.74	13.13	15.66	13.86	13.11	12.53
nnn	61.7	77.8	68.0	61.99	53.2	70.36	61.1	56.73
	13.32	15.15	12.73	13.53	13.90	15.07	12.49	11.76
nfn	64.93	64.1	67.8	77.63	56.16	80.0	78.76	69.33
	12.23	15.20	12.45	14.59	11.01	14.90	14.70	13.90
ltn	68.33	59.2	62.4	62.3	65.23	64.63	64.56	65.03
	19.23	26.79	24.96	19.37	19.22	18.58	18.80	19.31
ltc	68.83	57.1	65.43	65.00	66.33	72.1	65.23	69.0
	19.47	25.77	26.82	22.45	19.16	19.98	17.87	21.07
lnc	72.76	59.2	68.0	63.13	66.5	71.03	65.56	68.16
	19.85	27.38	27.00	21.10	19.21	20.17	18.66	20.7
dtc	51.33	56.50	65.2	54.39	61.56	62.16	59.66	54.23
	10.10	10.93	13.39	11.00	13.49	11.76	12.64	11.51

Верхнее число в ячейке показывает $recall = \frac{\|A \cap B\|}{\|A\|} \times 100\%$, нижнее

— $precision = \frac{\|A \cap B\|}{\|B\|} \times 100\%$, где А — множество правильных ответов, а

В — множество ответов, полученных системой-автоответчиком на основе автоматически структурированной предметной области.

4. ИНДЕКСИРОВАНИЕ ДОКУМЕНТОВ И РАБОТА ЕСТЕСТВЕННО-ЯЗЫКОВОГО ИНТЕРФЕЙСА

При индексировании документов и разборе естественно-языковых запросов мы различаем три типа слов: значимые, незначимые и вспомогательные. Под незначимыми словами мы понимаем множество слов, которое может встретиться в любой предметной области, и таким образом, не несет конкретной фактической информации. К этим словам могут относиться вспомогательные глаголы, артикли, союзы, предлоги, местоимения, такие как «is», «and», «to», «yours». Вспомогательные слова — это те слова, которые сами по себе не несут конкретной информации, но помогают установить значительные для понимания смысла запроса связи между значимыми словами предметной области. К ним относятся, например, слова-отрицания, меняющие смысл сказанного с точностью до наоборот, такие как «no», «don't», «never». В некоторых случаях к важным вспомогательным словам могут относиться некоторые местоимения и предлоги (например, когда системе принципиально понять «кто кому что дал»). Все остальные слова, лежащие в предметной области, являются значимыми, и именно ими производится индексирование документов.

На рис. 3 естественно-языковой интерфейс показан в работе: пример запроса (query), извлеченная из него значимая лексика (retrieved items), ответ (answer) и узлы древнейшей иерархии, содержащие ответ.

The image displays a natural language interface with four main components:

- Query:** "I've recently bought a new car. What would you recommended about buying insurance?"
- Retrieved items:** "new car, buying, insurance"
- Answer:** "Things you should consider when purchasing automobile insurance include: Decide how much liability coverage you want to carry. This is highly subjective. The liability levels you have on your other policies can serve as a guideline. Consult a financial professional if you need more advice. Determine which optional coverage you will need to feel protected. For example..."
- Hierarchical tree:** A tree structure under "Subject domain" showing categories like "Personal Financial Services", "Automobile Insurance", "Features", and "Moving". Specific items like "Buying auto insurance" and "New car" are highlighted in blue.

Рис. 3. Функциональность естественно-языкового интерфейса

В данной реализации системы-автоответчика индексирование также производится автоматически. Следует заметить, что присутствующие в индексах категорий значимые слова не записываются физически в индексы всех подкатегорий этих категорий по глубине иерархии, но наследование содержания индексов от категории к её подкатегории учитывается алгоритмом информационного поиска естественно-языкового интерфейса. Например, категории «страхование транспортных средств» и «страхование жизни» будут непосредственными подкатегориями категории «страхование». Так как все подкатегории категории «страхование» наследуют индекс категории «страхование» по глубине иерархии, нет необходимости хранить в них индекс категории «страхование». Таким образом, индексы на уровне категорий «страхование транспортных средств» и «страхование жизни» будут содержать слова близкие к таким, как «автомобиль», «жизнь» и прочим терминам, характеризующим именно эти категории. Используемые правила для алгоритма нахождения наиболее соответствующих запросу категорий дерева были подробно описаны в предыдущей работе [8].

5. РЕЗУЛЬТАТЫ

В рамках предложенного нами статистического подхода к построению иерархического представления предметной области были проведены эксперименты по структурированию предметной области страхования транспортных средств и страхования жизни (83 категории). Установлено, что автоматически сгенерированные иерархии с отношением «от простого к сложному» не совпадают в смысле совмещения наложением с иерархиями с отношением «is-a» и «part-of», построенными вручную (что вполне естественно в силу различия отношений упорядочивания). Как правило, в автоматически сгенерированных иерархиях мы наблюдали меньшее количество уровней, но структура такой иерархии существенно зависит от качества используемых текстовых документов, выбора функции веса и меры близости. Ниже приведен фрагмент автоматически сгенерированного дерева для предметной области Life Insurance (LI). Здесь номера (0, 1, 2,...) являются уровнями иерархии, а текстовые значения (Decreasing TLI, Graded-premium WLI,...) – названиями категорий. Расположение записей типа номер-название отображают отношения зависимости между категориями (зависимое множество подкатегорий находится непосредственно под своей категорией).

- 0 : Decreasing TLI
 - 1 : Graded-premium WLI
 - 2 : Rising premiums
 - 2 : Modified-premium WLI
 - 1 : Increasing TLI
 - 1 : Funeral insurance
 - 2 : Medical Payments
 - 1 : Continuous-premium WLI
 - 2 : Limited-payment WLI
 - 1 : Single-premium WLI
 - 2 : Indeterminate premium WLI
 - 2 : Current assumption WLI
 - 3 : Universal WLI

Мы оцениваем качество автоматического структурирования предметной области, исходя из результатов работы системы-автоответчика, построенной на основе этой области. Иными словами, на каждый переданный запрос системой-автоответчиком возвращалось два множества документов-ответов *A* и *B*. Первое из них получалось в результате работы системы с использованием области, структурированной вручную лингвистом. Это множество принималось за множество правильных ответов. Множество ответов *B* — результат работы системы с использованием автоматически структурированной области. Оценки работы системы-автоответчика, в зависимости от функций веса документа и мер сходности двух документов, усредненные по множеству тестовых запросов и полученные в широко известных терминах *precision* и *recall* [5], приведены в табл. 1. Используемые в тестировании функции веса документа [6] и меры семантической близости документов [1] показаны на рис. 4 и 5 соответственно.

Одной из основных проблем систем поиска и автоматического ответа является низкое значение *precision*, поэтому относительно небольшое значение *precision* в наших экспериментах является «нормальным». Результаты показывают, что при использовании отдельных функций веса и мер близости, значение *recall* превышает 75%. Принимая во внимание временные и умственные затраты на ручное структурирование информации, мы полагаем, что наш подход очень перспективен.

(CD) Camberra Distance	$sim(X, Y) = \sum_{1 \leq i \leq \ F\ } \frac{ x_i - y_i }{(x_i + y_i)}$
(CC) Correlation Coefficient	$sim(X, Y) = \left(\frac{1-r}{2} \right)^{\frac{1}{2}}, r = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sqrt{(\sum x_i^2 - \frac{(\sum x_i)^2}{n})(\sum y_i^2 - \frac{(\sum y_i)^2}{n})}}$
(JA) Jaccard Association	$sim(X, Y) = \left(1 - \frac{a}{a+b+c} \right)^{-1},$ где $a = \ X \cap Y\ $, $b = \ X \setminus Y\ $, $c = \ Y \setminus X\ $
(DP) Dot Product	$sim(X, Y) = \sum_{1 \leq i \leq \ F\ } x_i y_i$
(SP) Simple Intersection	$sim(X, Y) = a$
(SDA) Sorensen Dice Association	$sim(X, Y) = \left(1 - \frac{a+d}{2a(2a+b+c)} \right)^{-1},$ где $d = \ F \setminus (X \cup Y)\ $
(TD) Taxonomic Distance	$sim(X, Y) = \left(\sum_{1 \leq i \leq \ F\ } (x_i - y_i)^2 \right)^{\frac{1}{2}}$
(Cosine) Cosine Measure	$sim(X, Y) = \frac{\sum_{1 \leq i \leq \ F\ } x_i y_i}{\sqrt{\sum_{1 \leq i \leq \ F\ } x_i^2 \sum_{1 \leq i \leq \ F\ } y_i^2}}$

Рис. 4. Функции меры семантической близости документов. $\{x_i\}$, $\{y_i\}$ — частоты слов документов X и Y соответственно. F — множество слов предметной области

(entr) Entropy Weight	$w(X) = \frac{-\sum \ln(x_i)}{\ X\ }$	(stat) Statistical Weight	$w(X) = \prod \frac{1}{x_i}$
(atn)	$w(X) = \sum x_i^* \left(\frac{1}{2} + \frac{1}{2} \frac{x_i}{\max x_i} \right)$	(bnn)	$w(X) = \ X\ $
(ntc)	$w(X) = \sum \frac{x_i x_i^*}{\sqrt{\sum (x_i x_i^*)^2}}$	(npr)	$w(X) = \sum \left(x_i \ln \left(\frac{n - x_i^*}{x_i^*} \right) \right)$
(nnn)	$w(X) = \sum x_i$	(nfn)	$w(X) = \sum \ln \left(\frac{n}{x_i^*} \right)$
(ltm)	$w(X) = \sum \{ (\ln(x_i) + 1) x_i^* \}$	(lte)	$w(X) = \sum \frac{(\ln(x_i) + 1) x_i^*}{\sqrt{\sum ((\ln(x_i) + 1) x_i^*)^2}}$
(lnc)	$w(X) = \sum \frac{\ln(x_i) + 1}{\sqrt{(\ln(x_i) + 1)^2}}$	(dte)	$w(X) = \sum \frac{(\ln(\ln(x_i) + 1) + 1) x_i^*}{\{ (\ln(\ln(x_i) + 1) + 1) x_i^* \}^2}$

Рис. 5. Функции веса документа. Здесь x_i обозначает частоту i -го слова внутри документа X , а x_i^* — частоту этого слова во всей предметной области

СПИСОК ЛИТЕРАТУРЫ

1. **Anquetil N., Fourier C., Lethbridge T.** Experiments with Hierarchical Clustering Algorithms as Software Remodularization Methods — Ottawa, 1999. — (Tech. Rep. / Department of Computer Science. University of Ottawa; N 99-1).
2. Approach to Development of a System for Speech Interaction with an Intelligent Robot / **Cheblakov, G.B., Dinenberg, F.G., Levin, D.Ya., Popov, I.G., Zagorulko, Yu.A.** // Perspectives of System Informatics: Proc / Conf. held in Novosibirsk, Russia, 1999 / Ed. by D. Bjørner *et al.* — Berlin etc.: Springer, 1999. — P. 517–529. — (Lect. Notes Comput. Sci.; 1755).
3. **Dinenberg, F. G., Levin, D. Ya.** Natural Language Interfaces for Environmental Data Bases // Applications of Natural Language to Information Systems. — Amsterdam etc.: IOS Press, 1996.
4. **Freeman, R., Yin, H.** Self-Organising Maps for Hierarchical Tree View Document Clustering Using Contextual Information // Lect. Notes Comput. Sci.— Berlin etc., 2002.— Vol. 2412.— P. 123–128.

5. **Manning, C.D., Schütze, H.** Foundations of Statistical Natural Language Processing — Cambridge: The MIT Press, 2001.
6. **Savoy J.** Cross-language information retrieval: experiments based on CLEF 2000 corpora // Information Processing and Management. — 2003. — Vol. 39. — P. 75–115
7. **Vinokourov, A., Girolami, M. A.** Probabilistic Framework for the Hierarchic Organisation and Classification of Document Collections // Intellig. Inform. Systems. — 2002. — Vol. 18. — P. 153–172
8. **Zhdanova, A.V., Shishkin, D.V.** Classification of E-mail Queries by Topic: Approach Based on Hierarchically Structured Subject Domain // Lect. Notes. Comput. Sci. — Berlin etc., 2002. — Vol. 2412. — P. 99–104.
9. **Zipf, G. K.** Human Behavior and the Principle of Least Effort — Cambridge: Addison-Wesley, 1949.

Лобив И.В. Мехонцев Д.Ю. Мурзин Ф.А.

**ВОССТАНОВЛЕНИЕ ПОЛОЖЕНИЯ ТЕЛА В ПРОСТРАНСТВЕ
ПО ДАННЫМ, ПОСТУПАЮЩИМ С ОДНОМЕРНЫХ КАМЕР
ДЛЯ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ
АНАЛИЗА ДВИЖЕНИЯ ОБЪЕКТОВ**

ПОСТАНОВКА ЗАДАЧИ

В пространстве имеется твердое тело с известной геометрией, на котором в определенных местах закреплены светоизлучающие диоды, в дальнейшем называемые маркерами, которым присвоены уникальные номера. Также имеется набор одномерных детекторов, расположенных вокруг твердого тела, которые достаточно часто (200–300 раз в сек.) фиксируют положение маркеров. Каждый маркер обладает уникальными характеристиками (цвет, частота мерцания, и т.п.), по которым детектор может отличить его от других маркеров и восстановить его номер. В силу одномерности детектора, о каждом маркере, попадающем в его поле видимости, можно получить лишь уравнение плоскости, в которой этот маркер содержится. Все плоскости, получаемые от фиксированного детектора, пересекаются по одной прямой (“фокусная прямая”).

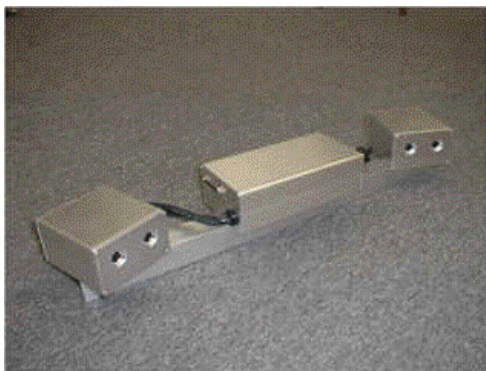


Рис. 1. Блок из четырех одномерных камер

Задача состоит в том, чтобы по имеющимся данным в реальном времени восстанавливать положение твердого тела с максимальной точностью.

На рис.1 показан блок из четырех одномерных камер. Таких блоков, расположенных в разных частях пространства, может быть несколько.

Осложняющие факторы

- В нескольких последовательных кадрах может быть недостаточно информации для однозначного восстановления положения тела (недоопределенность).
- В силу несовершенства детектора возможна ситуация, когда он выдает ошибочный номер маркера.

Математическая постановка

Положение тела в пространстве можно представить в виде $Ax + b$, где $b = (u, v, w)$ — вектор сдвига, A — ортогональная матрица поворота тела относительно начального положения.

Задача будет решена, если будут найдены величины A, b , при которых отклонение положения маркеров от соответствующих плоскостей будут минимальны в некоторой норме.

Научная новизна

Абсолютным лидером в данной области является фирма Vicon [5]. Ее продукты использовались для съемок таких знаменитых фильмов, как: «Titanic», «Gladiator», «The Mummy», «The Mummy Return», «Star Wars Episode 1 — the Phantom Menace», «The Patriot», «Enemy at the Gates», «Pearl Harbor» и др. Также ее продукты используются для постановки физических экспериментов, производства игр, обучения.

Рис. 2, 3 иллюстрируют работу продуктов разных фирм.

Попробуем выявить одну из главных проблем, которая возникает у исследователей [5, 6] в данной области. Подсчитаем объем информации, который должен обрабатываться в реальном времени.

Рассмотрим профессиональную 2d камеру с обычными характеристиками (1024x1024 True Color). Требование на рабочую частоту камеры — 300 фреймов в секунду (fps). Это требование вытекает из особенности предметной области. Частота промышленных мониторов и телеприемников — 24–30 fps, т.е. для того чтобы итоговое изображение на экране было непре-

рывным, частота на этапе обработки должна быть в 10 раз больше, т. е. 300 fps. Получается видеопоток 900Mb в секунду с каждой камеры. Поскольку камер 4 (например, в одной из конфигураций системы Vicon 8i), то в итоге получается более **3Gb** в секунду.



Рис. 2. Вид продукта лидера в данной области фирмы Vicon



Рис. 3. Сцена для работы продукта фирмы www.motionanalysisstudios.com

Этот показатель является огромным для сегодняшнего уровня развития вычислительной техники как для передачи такого объема информации в реальном времени, так и для обработки. Поэтому все направления исследований направлены на уменьшение размера информационного потока. Например, применение методов сжатия, увеличение количества вычислительных узлов и их мощности, привлечение суперкомпьютеров.

Мы предлагаем принципиально новое решение этого вопроса. Вычислим то же самое для одномерных камер. Размер изображения средней камеры 1024x1, условие на частоту то же самое 300 fps. Получаем с одной камеры 900Kb в секунду.

Так как камер требуется больше, то возьмем порядка 40 камер. Итого около **40Mb** в секунду, этот показатель уже вполне реален как для обработки, так и для передачи.

ОПИСАНИЕ МЕТОДА

Основная идея алгоритма заключается в том, чтобы в начале отфильтровать входные данные, а затем по ним восстановить положение тела [7]. При этом недостающая информация о положении тела дополняется данными с предыдущих кадров на основе интерполяции.

1. Случай, когда на одном временном шаге данных (плоскостей) достаточно для однозначного восстановления положения тела

Матрица A задается направлением (p, q, r) , относительно которого происходит поворот тела, и синусом угла поворота $\sin \phi$. Заметим, что вектор (p, q, r) одновременно может задавать и угол поворота по формуле $p^2 + q^2 + r^2 = \sin^2 \phi$, (мы используем модуль вектора для характеристики угла). Таким образом, матрица A задается тремя числами, и это представление есть непрерывное соответствие между ортогональными матрицами и единичным шаром в трехмерном пространстве.

Все это позволяет нам для нахождения преобразования $Ax + b$ воспользоваться методом наименьших квадратов, который в случае достаточности данных позволяет сразу по исходным данным (уравнениям плоскостей) восстановить положение тела, минуя определение положения каждого маркера по отдельности (что зачастую невозможно, потому что на каждый маркер может приходиться менее чем три плоскости).

Например, в случае если на одном шаге времени видны одновременно 3 маркера для восстановления положения тела, достаточно знать по две плоскости на каждый маркер (итого 6 плоскостей) или 1 плоскость про первый маркер, 2 плоскости про второй, 3 плоскости про третий маркер (также 6 плоскостей).

Метод наименьших квадратов позволяет также использовать избыточное количество плоскостей (более 6) для повышения точности восстановления. Пример показан на рис. 4.

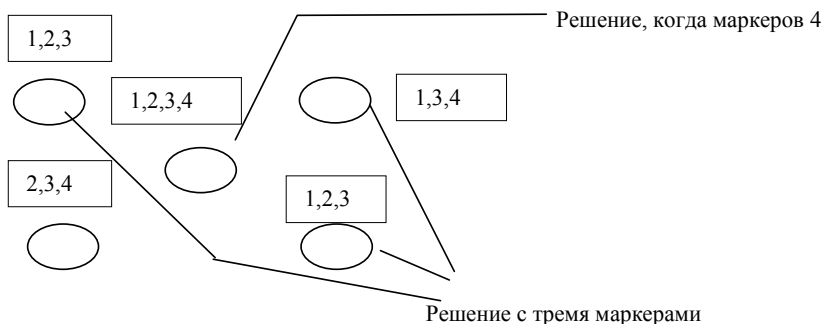


Рис. 4. Эффект увеличения точности восстановления положения при использовании информации с большего количества детекторов

Выпишем в явном виде представление матрицы A :

$$A = A(p, q, r) = \begin{pmatrix} c & -r & q \\ r & c & -p \\ -q & p & c \end{pmatrix} + \frac{1}{1+c} \begin{pmatrix} p^2 & pq & pr \\ qp & q^2 & qr \\ rp & rq & r^2 \end{pmatrix},$$

где $c^2 = 1 - (p^2 + q^2 + r^2) = \cos^2 \phi$.

Пусть на данном временном шаге известны N плоскостей, заданных в виде $(n_i, X) = d_i$ $i = 1 \dots N$, где $n_i = (a_i, b_i, c_i)$ — вектор нормали к плоскости, (\circ, \circ) — скалярное произведение, а также N точек $X_i = (x_i, y_i, z_i)$, характеризующих координаты маркера в системе координат твердого тела (предполагается, что эта информация считывается из файла с описанием твердого тела).

Метод наименьших квадратов приводит к следующей задаче:

$$\sum_{i=1}^N [(n_i, AX_i + b) - d_i]^2 \rightarrow \min,$$

Взяв частные производные соответственно по p, q, r, u, v, w , приходим к следующей системе уравнений:

$$\left\{ \begin{array}{l} \sum_{i=1}^N \left(\frac{\partial A}{\partial p} X_i, n_i \right) ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N \left(\frac{\partial A}{\partial q} X_i, n_i \right) ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N \left(\frac{\partial A}{\partial r} X_i, n_i \right) ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N a_i ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N b_i ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N c_i ((AX_i + b, n_i) - d_i) = 0. \end{array} \right.$$

В полученную систему неизвестные u, v, w входят линейно, поэтому мы сразу можем их исключить (выразить через p, q, r) линейной заменой:

$$b = S^{-1} \sum_{i=1}^N d_i n_i - \sum_{i=1}^N S^{-1} S_i A X_i, \text{ где}$$

$$S_i = \begin{pmatrix} a_i^2 & a_i b_i & a_i c_i \\ b_i a_i & b_i^2 & b_i c_i \\ c_i a_i & c_i b_i & c_i^2 \end{pmatrix}, \quad S = S_1 + S_2 + \dots + S_N.$$

В итоге у нас остается система трех нелинейных уравнений с тремя неизвестными p, q, r :

$$\begin{cases} f_1(p, q, r) = 0 \\ f_2(p, q, r) = 0 \\ f_3(p, q, r) = 0 \end{cases}$$

или в векторном виде

$$F(P) = 0, \text{ где } F = (f_1, f_2, f_3), \quad P = (p, q, r).$$

Для ее решения воспользуемся методом последовательных приближений Ньютона [3].

Пусть на n -том шаге известно приближение $P_n = (p_n, q_n, r_n)$, тогда следующее приближение P_{n+1} находится по формуле: $P_{n+1} = P_n - J^{-1}F(P_n)$, где J — матрица Якоби преобразования F ,

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial p} & \frac{\partial f_1}{\partial q} & \frac{\partial f_1}{\partial r} \\ \frac{\partial f_2}{\partial p} & \frac{\partial f_2}{\partial q} & \frac{\partial f_2}{\partial r} \\ \frac{\partial f_3}{\partial p} & \frac{\partial f_3}{\partial q} & \frac{\partial f_3}{\partial r} \end{pmatrix}.$$

Как известно, метод Ньютона очень быстро сходится в случае хорошего начального приближения.

В нашем алгоритме для начального приближения используются данные, полученные в результате экстраполяции по предыдущим временным шагам (с предыдущих кадров) — это обеспечивает хорошее начальное приближение и позволяет обойтись в методе Ньютона двумя-тремя итерациями для достижения нужной нам точности.

2. Случай, когда на одном временном шаге нам приходит недостаточное количество плоскостей (менее 6), для того чтобы однозначно восстановить положение тела

Очевидно, что в этом случае искомые величины p, q, r, u, v, w также удовлетворяют раннее описанной системе уравнений $F(P) = 0$, но решений эта система имеет много (как правило, имеется целое пространство решений).

Недостающие плоскости мы получаем с помощью экстраполяции по предыдущим временным шагам следующим способом.

На первом этапе вычисляются положения невидимых в данный момент точек на основе того, что мы знали их положение (а следовательно, приближенно знали вектор скорости и ускорения) на предыдущих шагах [4].

На втором этапе через вычисленные точки проводятся плоскости, которыми и дополняются исходные неполные данные.

Более точно, пусть (в таких же обозначениях, как и ранее) на данном временном шаге известны M дополнительных (полученных на основе интерполяции) плоскостей, заданных в виде $(n_i, X) = d_i$, $i = N + 1 \dots N + M$, а также M им соответствующих точек $X_i = (x_i, y_i, z_i)$.

Тогда в предположении, что $M + N \geq 6$, возникает задача [8]

$$\sum_{i=1}^{N+M} [(n_i, AX_i + b) - d_i]^2 \rightarrow \min,$$

полностью аналогичная предыдущему случаю.

Заметим, что этим методом можно вычислять более точное положение тела на предыдущих временных шагах, используя информацию, полученную на последующих шагах (направить метод в “прошлое”).

3. Фильтрация шумов

В силу несовершенства детекторов иногда фиксируются неверные плоскости (т.е. маркер не лежит в плоскости, а находится вдали от нее), например, если перепутается номер маркера.

Для борьбы с таким видом шумов положение тела восстанавливается дважды. Первый раз восстановление происходит по всем имеющимся плоскостям (в том числе и неверным).

Затем мы проверяем, насколько каждая плоскость отклоняется от вычисленного положения, и при втором восстановлении не учитываются те плоскости, в которых это отклонение велико (например, в 3 раза превосходит среднее отклонение и т.п.).

ОПИСАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

Была написана реализация вышеизложенного алгоритма на языке C++ [1,2]. В виде dll-библиотеки для Windows [9,10] и в виде запускаемого файла для Linux. Программа тестировалась на данных, полученных с реальной системы, и показала высокую точность и устойчивость к шумам. Среднее время обработки — 10000 временных шагов менее 1 сек.

СПИСОК ЛИТЕРАТУРЫ

1. **Gamma E., Helm R., Johnson R., Vlissides J.** Design Patterns. — Addison Wesley Longman Inc, 1995 — 132 p.
2. **Booch G.** Object-oriented analysis and design. 2-nd ed. — Addison Wesley Longman Inc, 1994 — 187 p.
3. **Самарский А.А.** Введение в численные методы. — М.: Наука, 1982. — 272 с.
4. **Рябенский В.С.** Введение в вычислительную математику. Учебное пособие для вузов. — М.: Физматлит, 1994. — 336 с.

5. **Vicon Inc.** <http://www.vicon.com>
6. **Motion Analysis Studio Inc.** <http://www.motionanalysisstudios.com>
7. **Мехонтцев Д.Ю., Лобив И.В., Селезнев К.С.** Слежение и определение скорости движущихся на плоскости объектов в реальном времени. // Современные проблемы конструирования программ. — Новосибирск, 2002. — С. 243–246.
8. **Коновалов А.Н.** Введение в вычислительные методы линейной алгебры. — Новосибирск: ВО «Наука», 1993. — 59 с.
9. **Codeguru** <http://www.codeguru.com>
10. **Code Project** <http://www.codeproject.com>

Рыжов В. С.

ПОСТРОЕНИЕ РАСПРЕДЕЛЕННЫХ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ИНТЕГРИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПРЕДПРИЯТИЯ

ВВЕДЕНИЕ

В данной работе рассматривается модель многоуровневой информационной системы, интегрирующей разнородные классы объектов различной конфигурации и степени сложности. Предложено архитектурное решение, позволяющее динамически расширять действующую систему, подключая к ней объекты как уже имеющихся в ней классов, так и объекты новых классов. В работе также обобщается опыт практической реализации модели в виде распределенного приложения. Анализируется опыт эксплуатации разработанных систем, включающих в себя подсистемы контроля доступа на объект, энергоснабжения, пожарной безопасности, видеонаблюдения и других систем жизнеобеспечения. Работа затрагивает следующие аспекты построения систем обеспечения безопасности и жизнедеятельности предприятия.

1. Интеграция разнородных подсистем в единую систему.
2. Адаптация системы к объектам различной конфигурации и степени сложности.
3. Защита самой информационной системы на всех уровнях в целях обеспечения безопасности предприятия.

Необходимость интеграции разнородных подсистем обусловлена тем, что в систему безопасности предприятия входят подсистемы контроля доступа, видеонаблюдения, пожарной охраны и другие системы жизнеобеспечения, т. е. совершенно разнородные системы, включающие в себя объекты различных типов, но их существование в едином информационном пространстве требует тесной интеграции.

Требование легкости адаптации системы обусловлено существенными различиями объектов, жизнедеятельность которых должна обеспечить система. Необходимость множественного внедрения такой системы вытекает из желательной унификации поведения систем жизнеобеспечения пред-

приятия и немалой стоимости разработки таких систем. Как следствие, становится необходимой возможность внесения изменений в сценарии поведения системы во время ее функционирования.

Особенно важен аспект безопасности самой системы, призванной обеспечивать безопасность предприятия. Под безопасностью понимается как обеспечение защиты системы на разных уровнях, например, защиты от несанкционированного доступа и прослушивания сети, так и защиты от физического повреждения данных, а также способность системы к восстановлению в случае всех видов повреждений.

ПРЕДЛОЖЕНИЯ ПО АРХИТЕКТУРЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ И ЖИЗНЕДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ

Для правильного выбора архитектуры и способов реализации системы необходимо в полной мере сформулировать требования к системе.

Стабильность работы

Каждая часть системы должна стабильно работать, максимально независимо от остальных частей системы.

В частности, предлагаемая архитектура должна предусматривать решение следующих проблем.

- Работоспособность аппаратных средств (контроль и восстановление после сбоев).
- Подтверждение доставки сообщений при сетевом взаимодействии или гарантированной доставки таких сообщений (буферизация потоков данных).
- Работоспособность сценариев обработки, создаваемых пользователем (контроля времени обработки и зависания).

Планирование межзадачного взаимодействия

Система управления потоками работ должна обеспечивать спецификацию и планирование межзадачных зависимостей. Кроме того, параллельным выполнением могут поддерживаться восстановление и управление; в этом случае возможна интеграция планировщика, поддерживающего со-

блюдение межзадачных зависимостей, с системой управления ослабленными транзакциями.

Система управления потоком работ состоит из планировщика и агентов задач. Агент задачи контролирует выполнение задачи обрабатываемым объектом; для каждой задачи существует один агент. Планировщик — это программа, которая обрабатывает потоки работ, запуская на выполнение различные задачи, обрабатывая различные события и оценивая условия, связанные с межзадачными зависимостями. Планировщик может представить задачу на выполнение (агенту задачи) или потребовать прекращения выполнения ранее запущенных задач. В случае транзакции, затрагивающей несколько баз данных, задачи являются подтранзакциями, а обрабатываемые объекты — это локальные СУБД. В соответствии со спецификацией потока работ планировщик обеспечивает планирование зависимостей и является ответственным за обеспечение того, что все задачи достигнут приемлемого состояния завершения.

Существует три архитектурных подхода к построению систем управления потоками работ. При использовании централизованного подхода имеется один планировщик, который планирует задачи всех параллельно выполняющихся потоков работ. Частично распределенный подход предполагает наличие одного (экземпляра) планировщика для каждого потока работ. В случае, когда вопросы параллельного выполнения могут быть отделены от функции планирования, последний подход является более естественным. В случае полностью распределенного подхода планировщик отсутствует; агенты задач координируют свое выполнение, взаимодействуя друг с другом для разрешения межзадачных зависимостей и прочих требований выполнения потоков работ.

Восстановление потока работ

Целью восстановления после отказа при выполнении потоков работ является обеспечение атомарности потоков работ в связи со сбоями. Процедуры восстановления должны гарантировать, что в случае возникновения отказа в любом из компонентов, принимающих участие в обработке потока работ (включая планировщик), поток все равно достигнет приемлемого состояния — возможно, с использованием компенсации. Мы можем предположить, что обрабатываемые объекты, принимающие участие в выполнении потока работ, обладают своими собственными локальными системами восстановления и сами обрабатывают возникающие в них локальные отказы. Поэтому мы будем обсуждать только отказы менеджеров выполне-

ния потока работ (планировщиков и менеджеров параллельного выполнения).

Чтобы восстановить контекст среды выполнения, процедура восстановления после отказа должна восстановить информацию о состоянии к моменту отказа, включая информацию о состоянии выполнения каждой задачи и информацию о зависимостях планирования для менеджера параллельного выполнения. Следовательно, соответствующая статусная информация должна сохраняться на надежном запоминающем устройстве [2].

Отказ менеджера параллельного выполнения может быть обнаружен посредством механизмов временного контроля, а информация о состоянии может быть реконструирована путем просмотра журналов. В предположении возможности прямого восстановления [3] и идемпотентности обрабатываемых объектов, выполнение потока работ может быть продолжено простым перезапуском незавершенных задач.

Нам также требуется обращать внимание на содержимое очередей запросов. Если используется механизм, обеспечивающий функции стабильных программных каналов (например, в системе VMS компании DEC очереди могут быть реализованы с использованием почтовых ящиков — стабильных очередей сообщений, доступных как виртуальные устройства ввода/вывода), сообщения хранятся на постоянном запоминающем устройстве и не теряются в случае отказа. Если очереди хранятся на ненадежном запоминающем устройстве (например, в среде ОС UNIX очереди могут быть реализованы с использованием сокетов), вместо восстановления содержимого очередей можно пойти на то, чтобы планировщики повторно послали свои запросы.

Во многих коммерческих продуктах для распределения задач по обрабатывающим станциям, ответственным за их выполнение, используются интерфейсы электронной почты. Это простое решение обладает преимуществами, вытекающими из относительной надежности существующих систем электронной почты и содержащихся в них возможностей по организации очередей и повторных посылок сообщений.

Планирование потока работ

Основными целями планировщика является обеспечение следующих требований.

- **Корректность планирования.** Процесс планирования не может нарушить ни одну из зависимостей, представленную в спецификации потока работ. Кроме того, планировщик связан ограничения-

ми, налагаемыми глобальным управлением параллельным выполнением, поскольку неконтролируемое взаимодействие задач, принадлежащих разным потокам работ, может привести к неверным результатам. Определение того, могут ли быть удовлетворены временные зависимости планирования, является особенно трудной задачей [1]. Планировщик должен учитывать, что при наличии временных зависимостей логические значения предикатов планирования могут изменяться динамически, без каких бы то ни было воздействий со стороны системы. В то же время, эти зависимости ограничивают возможные действия планировщика (например, указывая, что задача не может быть запущена ранее 10:00).

- **Безопасность.** Планировщик должен гарантировать, что поток работ будет завершен в одном из специфицированных приемлемых заключительных состояний. Перед попыткой выполнения потока работ планировщик должен обратиться к спецификации, чтобы проверить, не завершается ли этот поток в неприемлемом состоянии. Если планировщик не может гарантировать, что поток работ будет завершен в приемлемом состоянии, он должен отвергнуть такую спецификацию, не пытаясь выполнить данный поток работ. Даже если спецификация потока работ является безопасной, т. е. приемлемое состояние завершения может быть всегда достигнуто, все равно остаются проблемы с выполнением потока работ, возникающие из-за возможности возникновения разного рода тупиков. Чтобы гарантировать отсутствие тупиков в потоке работ, можно использовать формальные методы спецификации и анализа. Задачей планировщика потока работ является построение стратегии выполнения, гарантирующей, что поток работ не будет завершен в неприемлемом состоянии.
- **Оптимальная политика планирования.** Планировщику следует достигать приемлемых состояний завершения «оптимальным» образом. Тем не менее, понятие «оптимальный» может различаться для разных приложений. Одним из возможных определений может быть достижение цели за наиболее короткое время. Другим — связывание оценочной функции с выполнением каждой задачи. Тогда целью планировщика было бы выполнение всего потока работ с минимальной стоимостью. Если заранее известны вероятности успешного выполнения задач, планировщик может использовать их

для нахождения стратегии выполнения, приводящей с максимальной вероятностью к глобальному успешному завершению.

- **Обработка отказов.** Планировщик должен быть способен достигать приемлемого состояния завершения даже в случае возникновения отказа. Например, планировщик мог бы продолжить обработку после отказа и соответствующего восстановления, как будто «ничего не произошло», обеспечивая «возможность прямого восстановления» (forward recoverability). И наоборот, планировщик мог бы прекратить выполнение всего потока работ (т. е. достигнуть одного из глобальных состояний аварийного завершения). Оба подхода требуют того, чтобы информация о состоянии сохранялась в случае отказа, поскольку даже во втором случае может потребоваться фиксация или возобновление выполнения некоторых подтранзакций (например, компенсирующих подтранзакций). Следовательно, планировщик должен сохранять на надежном запоминающем устройстве всю информацию о своем состоянии, которая может потребоваться для восстановления и продолжения обработки.

Быстродействие системы

Генерация статистических отчетов должна занимать приемлемое время. В системе должны быть предусмотрены механизмы подготовки отчетов по большому количеству информации, периодически (например, раз в день) и в фоновом режиме.

Время реакции системы на событие должно быть сопоставимо со временем реакции аппаратной части.

Средства конфигурирования системы

Система должна предлагать удобные средства конфигурирования, предназначенные для инсталлятора и наладчика системы. Следует предусмотреть возможность простого создания конфигурации для большого количества однотипных устройств, однотипных сценариев бизнес-логики.

Масштабируемость

Система должна предусматривать возможность реализации как минимальной конфигурации на одном компьютере, так и распределенной системы компьютеров, объединенных в сеть.

Развиваемость

Система должна предусматривать возможность расширения как путем добавления новых устройств, так и путем добавления новых компонентов бизнес-логики и вариантов пользовательского интерфейса.

Физически система представляет собой набор внешних устройств (электромеханических устройств охранной системы), подключенных к одному или нескольким компьютерам, объединенным в сеть Ethernet.

С программной точки зрения система представляет набор компонентов, взаимодействующих посредством обмена сообщениями. Компоненты системы функционируют в среде виртуальной операционной системы (виртуальной машины), которая не зависит ни от конкретной ОС, установленной на компьютере, ни от аппаратной реализации компьютера.

Предлагаемая архитектура позволяет строить системы различного масштаба — от минимальных, реализованных на одном компьютере с подключенными непосредственно к нему внешними устройствами, до распределенной по большому числу различных компьютеров, объединенных в единую сеть. Система строится по блочному принципу из компонентов, спектр которых может расширяться в процессе развития системы.

Ядро виртуальной ОС

Ядро виртуальной ОС реализует функции обработки и маршрутизации сообщений, обеспечивая взаимодействие между компонентами системы.

Каждый компонент системы использует API ядра системы для отправки сообщений. В случае локального адресата (в рамках одного узла виртуальной ОС) сообщение попадает в очередь на обработку соответствующего компонента. В противном случае сообщение попадает в очередь на отправку сообщения через сеть. Предполагается, что механизм отправки не предусматривает блокирования основного потока выполнения, т.е. сообщение ставится в очередь, а выполнение основного процесса продолжается. Алгоритмы доставки должны гарантировать доставку сообщений адресату. Сообщение удаляется из очереди только после подтверждения его передачи следующему звену в маршруте (при сетевой передаче) или при постановке

в очередь на обработку локальному адресату. При отсутствии подтверждения система периодически повторяет попытку доставки сообщения, пока не будет достигнуто предельное время передачи сообщения.

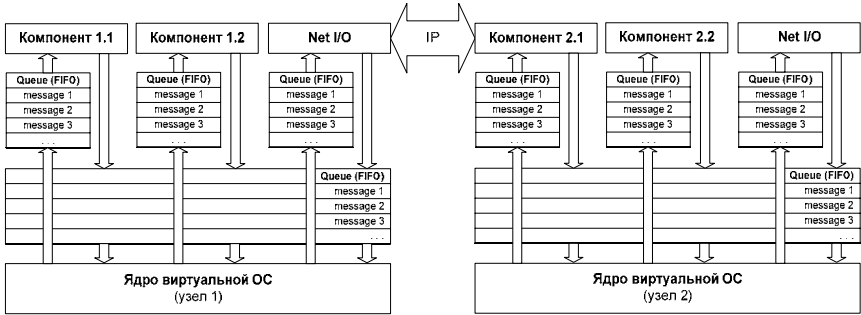


Рис. 1. Ядро виртуальной операционной системы – обработка сообщений

Механизм обработки очереди сообщений может быть различным. Базовым является механизм FIFO без распараллеливания процессов обработки: пока не окончена обработка текущего сообщения, остальные сообщения ждут в очереди. Возможно, что для некоторых компонентов необходимо будет реализовывать другие механизмы выборки из очереди (например, при обработке исключительных транзакций) и/или параллельной обработки сообщений.

Таким образом, ядро виртуальной системы является связующим элементом, который объединяет компоненты системы в виртуальную сеть, обеспечивающую обмен сообщениями.

Компоненты системы

При инициализации компонента система регистрирует имя (адрес) этого компонента, который затем используется при обмене сообщениями с другими компонентами системы. Для каждого компонента системы определен список сообщений (команд), которые он способен выполнить, а также список событий, которые могут быть сгенерированы данным компонентом. Кроме того, компонент системы может посылать сообщения другим компонентам.

Предполагается, что компоненты организованы в иерархию по принципу зависимости (например, vista → multiplexer → com-port), понятную для инсталлятора и наладчика системы. Эта иерархия должна быть отражена и на уровне GUI (Graphic User Interface).

Каждый компонент системы определяется алгоритмом обработки поступающих сообщений, параметрами конфигурации и внутренним состоянием. В общем виде взаимодействие с компонентом системы можно разделить на три категории: конфигурирование, контроль и управление. Предполагается, что помимо специфических команд, обрабатываемых определенным компонентом, любой компонент способен выполнять некоторые базовые команды (например, запрос о списке команд управления или параметров конфигурации).

Конфигурирование компонента

Для каждого типа компонентов предполагается наличие компонента, реализующего GUI для конфигурации, доступного инсталлятору системы. Такой компонент можно рассматривать как приложение, организующее пользовательский интерфейс и посылающее компоненту определенные сообщения с конфигурационными параметрами.

Список параметров конфигурации и соответствующих команд (сообщений) определен для каждого типа компонентов.

Контроль компонента

Для компонента определен список событий, которые могут быть сгенерированы им. Любой компонент системы может послать специальную команду данному компоненту и зарегистрировать себя как получателя оповещений о возникновении события. При возникновении указанного события все заинтересованные компоненты получают сообщение о возникновении события. Данная схема предназначена для организации терминалов и приложений, обрабатывающих и визуализирующих состояние компонентов. Для каждого типа компонентов предполагается наличие простейшего варианта GUI для контроля, доступного инсталлятору и наладчику системы.

Управление компонентом

Компонент может быть предназначен для управления внешним устройством или просто иметь некоторое внутреннее состояние, которым можно управлять. Для компонента определен список управляющих команд с необходимыми параметрами. Такие команды посылаются в результате обработки событий, реализованных в компонентах системы. Для каждого типа компонентов предполагается наличие простейшего варианта GUI для отправки команд управления, доступного инсталлятору и наладчику системы.

Менеджеры системы

Менеджеры системы — это абстрактный компонент системы, объединяющий компоненты по функциональным моделям. Менеджеры системы отвечают за инициализацию и конфигурирование компонентов при начальном запуске системы (включении питания), а также за сохранение и восстановление конфигурации системы.

Конфигурация менеджера содержит список и конфигурацию компонентов системы, а также список некоторых параметров, единых для компонентов менеджера. В некотором смысле, ядро узла операционной системы тоже является менеджером (а также и компонентом), но находится на самой верхушке иерархии компонентов.

Менеджер внешних устройств

Этот менеджер осуществляет загрузку (подключение) драйверов внешних устройств. Драйвер обрабатывает полученные от системы сообщения и генерируют соответствующие управляющие команды для устройства. В ответ на изменение состояния устройства (назовем такое изменение событием) драйвер генерирует сообщения и посылает их всем заинтересованным (зарегистрированным) получателям.

Драйверы устройств осуществляют функции контроля над работоспособностью устройств, а также инициализацию и восстановление после аппаратных сбоев.

Менеджер приложений

Приложения имеют API, аналогичный драйверам. Обычно приложения регистрируются у драйвера определенного устройства для обработки определенных событий, т.е. назначение приложений — осуществлять бизнес-логику обработки событий.

Одним из возможных приложений является компонент, позволяющий интерпретировать скрипты, написанные на некотором языке пользователем системы.

Другой вариант приложения — это компонент, организующий GUI для взаимодействия с другими компонентами системы.

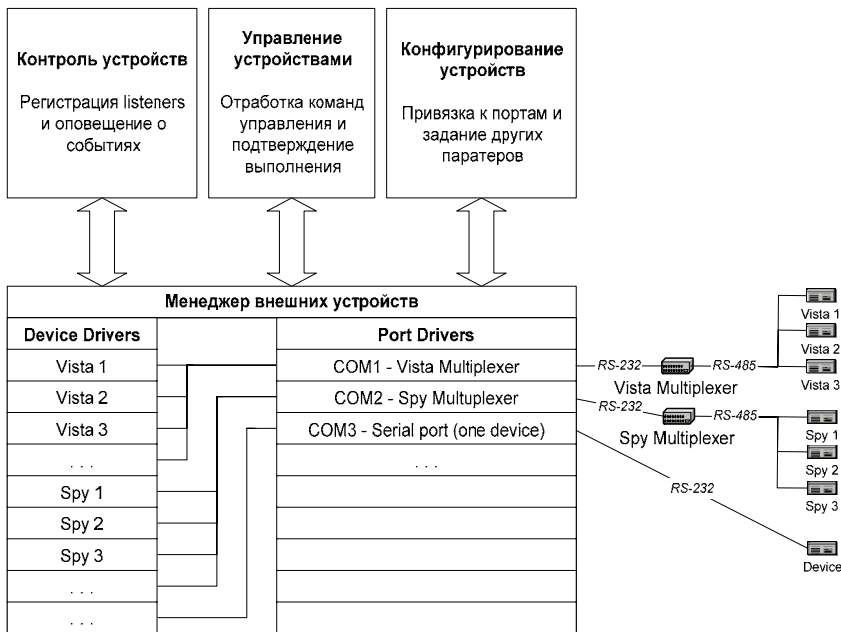


Рис. 2. Менеджер внешних устройств

Менеджер сетевого взаимодействия (Net I/O)

Этот менеджер осуществляет передачу сообщений другим узлам виртуальной ОС посредством шифрования протокола IP с обычной IP-маршрутизацией с помощью технологии SSL или IPSec, а также прием сообщений из сети для данной виртуальной машины с постановкой их в очередь на обработку. Кроме того, менеджер осуществляет защиту от несанкционированного доступа посредством механизма, подобного механизму firewall.

Менеджер данных

Этот менеджер взаимодействует с СУБД или другой системой хранения информации. Запросы к данным осуществляются через компоненты, аналогичные драйверам устройств, предоставляющие соответствующий функционал для различных типов данных. Таким образом, на уровне менеджера данных осуществляется отделение логики работы с данными от способа ее

хранения (структуры БД). На базе этого менеджера можно также создавать отдельные компоненты, предоставляющие функционал для генерации отчетов.

Менеджер синхронизации и управления транзакциями

Менеджер синхронизации и управления транзакциями позволяет создавать и использовать специальные объекты для синхронизации работы приложений и остальных компонентов. Соответствующие «драйвера-фабрики» предназначены для определенных типов объектов: семафоры (semaphores), флаги (mutex), затворы (locks) и т.д.

Менеджер предоставляет механизм создания и управления транзакциями. Компоненты системы используют данный менеджер для совместного исполнения последовательности действий в рамках одной транзакции.

РЕЗУЛЬТАТЫ ВНЕДРЕНИЯ

Первая версия описанной системы реализована в виде программного обеспечения аппаратно-программного комплекса «Восток». Опыт эксплуатации показал, что использование распределенной архитектуры позволяет значительно повысить отказоустойчивость системы, что является одним из главных требований, предъявляемых к системам безопасности.

Комплекс внедрен на ряде предприятий Сибири. АПК «Восток» дважды награждался Большой золотой медалью выставки «Сибирская ярмарка». В настоящее время ведутся работы над второй версией системы.

СПИСОК ЛИТЕРАТУРЫ

1. **Georgakopoulos D., Rusinkiewicz M., Litwin W.** Chronological Scheduling of Transactions with Temporal Dependencies. — Houston, 1991. — (Tech. Rep. / Dept. of Computer Science, University of Houston; UH-CS-91-03).
2. **Win W., Ness L., Rusinkiewicz M., Sheth A.** Concurrency Control and Recovery Multidatabase Work Flows in Telecommunication Applications // Proc. of the SIGMOD Conf., May 1993. — VLDB Journal. — 1994. — Vol. 3, N 1. — P. 1–28.
3. **Reuter A.** ConTracts: A Means for Extending Control Beyond Transaction Boundaries // Proc. of the 3rd Intl. Workshop on High Performance Transaction Systems, Asilomar, California, 1989.

СОДЕРЖАНИЕ

Предисловие	5
<i>Антонцева М. Ф.</i> Коалгебраическое исследование бисимуляционных параллельных процессов	7
<i>Аргиров В. С.</i> Трансляция SDL-спецификаций в модифицированные раскрашенные сети Петри.....	18
<i>Винокуров А. Н., Ильин И. В., Лобив И. В., Мурзин Ф. А., Половинко О. Н., Семич Д. Ф.</i> Программный комплекс, предназначенный для обработки результатов, полученных методом ядерного каротажа нефтяных скважин	23
<i>Грибовская Н. С.</i> Теоретико-категорная характеристика языковых эквивалентностей временных параллельных моделей.....	32
<i>Жданова А. В., Манкевич П. В.</i> Статический подход к семантическому структурированию предметных областей для запросов на естественном языке	42
<i>Лобив И. В., Мехонцев Д. Ю., Мурзин Ф. А.</i> Восстановление положения тела в пространстве по данным, поступающим с одномерных камер для системы реального времени анализа движения объектов	54
<i>Рыжов В. С.</i> Построение распределенных объектно-ориентированных интегрированных информационных систем обеспечения безопасности предприятия	63

CONTENTS

Preface	5
<i>Antontseva M. F.</i> Coalgebraic investigation of bisimulation parallel processes... 7	
<i>Argirov V. S.</i> Automatic constructing the net models of SDL specifications	18
<i>Vinokurov A.A., Ilyin I.V., Lobiv I.V., Murzin F.A., Polovinko O.N., Semich D.F.</i> The program complex for nuclear oil-wells carotage results processing.....	23
<i>Gribovskaya N. S.</i> The category theoretical characterization of the language equivalence for timed concurrent models.....	32
<i>Zhdanova A. V., Mankevich P. V.</i> Statistical approach to semantic structuring of hierarchical subject domain for natural language queries	42
<i>Lobiv I. V., Mekhontsev D. Yu., Murzin F. A.</i> Rigid body tracking for real-time system of object's motion analyses, data obtains from one-dimension optical detectors	54
<i>Ryzhov V. S.</i> Design and implementation of distributed object-oriented integrated information systems for providing building security...	63

- Коалгебраическое исследование бисимуляционных параллельных процессов / *Антонцева М. Ф.* // Тр. школы-конкурса молодых ученых ИСИ СО РАН «Теоретические и прикладные задачи информатики: новые подходы и решения». — Новосибирск, 2003. — С. 7–17.
- Цель данной работы — расширить два подхода, категориальный и коалгебраический, на неинтерливинговые модели, в частности, на представленные системами переходов с независимостью и помеченными структурами событий.
- Coalgebraic investigation of bisimulation parallel processes / *Antonitseva M. F.* // Proc. of school-and-contest of young researchers “Theoretical and applied problems of informatics: novel approaches and solutions”. — Novosibirsk, 2003. — P. 7–17.
- The aim of this paper is to extend coalgebra semantic and categorical methods to noninterleaving models, in particular, to transition systems with independence and labelled event structures.
- Трансляция SDL-спецификаций в модифицированные раскрашенные сети Петри / *Аргиров В. С.* // Тр. школы-конкурса молодых ученых ИСИ СО РАН «Теоретические и прикладные задачи информатики: новые подходы и решения». — Новосибирск, 2003. — С. 18–22.
- Описывается алгоритм автоматического перевода статического подмножества SDL в модификацию раскрашенных сетей Петри — иерархические временные типизированные сети (ИВТ-сети) с последующей генерацией сетевой модели в системе SPV (SDL Protocol Verifier). ИВТ-сети расширяют безопасные раскрашенные сети Петри посредством понятий времени (семантика Мерлина), приоритетов, а также специальных мест, представляющих очереди фишек.
- Automatic constructing the net models of SDL specifications / *Argirov V. S.* // Proc. of school-and-contest of young researchers “Theoretical and applied problems of informatics: novel approaches and solutions”. — Novosibirsk, 2003. — P. 18–22.
- This work is aimed to research the problem of automatic constructing the net models of SDL specifications. In the paper SDL specifications with timers and priorities are considered. These specifications allow to represent a considerable class of communication protocols. Coloured Petri nets extended by priorities and Merlin's time concepts are used as net models.
- Программный комплекс, предназначенный для обработки результатов, полученных методом ядерного каротажа нефтяных скважин / *Винокуров А.А., Ильин И.В., Лобив И.В., Мурзин Ф.А., Половинко О.Н., Семич Д.Ф.* // Тр. школы-конкурса молодых ученых ИСИ СО РАН «Теоретические и прикладные задачи информатики: новые подходы и решения». — Новосибирск, 2003. — С. 23–31.
- Описываются математические постановки нескольких задач, возникающих при автоматизации процесса ядерного каротажа, и указаны методы их решения.
- The program complex for nuclear oil-wells carotage results processing / *Vinokurov A. A., Ilyin I. V., Lobiv I. V., Murzin F. A., Polovinko O. N., Semich D. F.* // Proc. of school-and-contest of young researchers “Theoretical and applied problems of informatics: novel approaches and solutions”. — Novosibirsk, 2003. — P. 23–31.
- Mathematical formulations of several problems appearing at automation of the process of nuclear carotage are described and the methods of their solution are presented.
- Теоретико-категорная характеристика языковых эквивалентностей временных параллельных моделей / *Грибовская Н. С.* // Тр. школы-конкурса молодых ученых ИСИ СО РАН «Теоретические и прикладные задачи информатики: новые подходы и решения». — Новосибирск, 2003. — С. 32–41.
- Разработан подход к построению временной частично упорядоченной языковой эквивалентности в контексте моделей реального времени с семантикой “истинного парал-

лелизма” — временных структур событий. Исследуемая эквивалентность формулируется в терминах равенства временных частично-упорядоченных языков системы.

The category theoretical characterization of the language equivalence for timed concurrent models / *Gribovskaya N. S.* // Proc. of school-and-contest of young researchers “Theoretical and applied problems of informatics: novel approaches and solutions”. — Novosibirsk, 2003. — P. 32–41.

The timed extension of a language equivalence is developed in the setting of timed event structures. As a main result this equivalence is characterized on the category.

Статический подход к семантическому структурированию предметных областей для запросов на естественном языке / *Жданова А. В.*, *Манкевич П. В.* // Тр. школы-конкурса молодых ученых ИСИ СО РАН «Теоретические и прикладные задачи информатики: новые подходы и решения». — Новосибирск, 2003. — С. 42–53.

Представлен статистический подход к построению структуры (иерархии) предметной области. Данный подход ориентирован на использование полученной структуры для поиска информации на естественном языке.

Statistical Approach to Semantic Structuring of Hierarchical Subject Domain for Natural Language Queries / *Zhdanova A. V.*, *Mankevich P. V.* // Proc. of school-and-contest of young researchers “Theoretical and applied problems of informatics: novel approaches and solutions”. — Novosibirsk, 2003. — P. 42–53.

A new scalable statistical algorithm for automatic semantic structuring of subject domain is proposed. The described methods are specifically oriented on the tasks of question answering and ontology generating. The hierarchy is extracted from the electronic text written in an ordinary natural language, e.g., English. Our implementation is empowered by the natural language interface, which allows conducting experiments within real data. Hierarchy navigation and evaluation the user’s conversance with the subject domain from analyzing the query cope with redundancy, which is common in modern question answering systems, and provide the most comprehensive answers. The results indicate that automatic information retrieval benefits a lot from employing hierarchical subject domain. Furthermore, the proposed principles of automatic subject domain construction are very promising in effective web and email question answering based on the vast amounts of information available in the electronic form.

Восстановление положения тела в пространстве по данным, поступающим с одномерных камер для системы реального времени анализа движения объектов / *Лобив И. В.*, *Мехонцев Д. Ю.*, *Мурзин Ф. А.* // Тр. школы-конкурса молодых ученых ИСИ СО РАН «Теоретические и прикладные задачи информатики: новые подходы и решения». — Новосибирск, 2003. — С. 54–62.

На твердом теле неподвижно закреплены некоторые маркеры. В пространстве расположены одномерные камеры (детекторы). В поле видимости детекторов находится твердое тело с маркерами. Описан подход для определения положения твердого тела по данным, поступающим с одномерных детекторов. Рассмотрены случаи, возникающие при недоопределенности и переопределенности системы. Предложенный метод позволяет решить одну из основных проблем в данной области – большой информационный поток (около 3GB в секунду), который возникает между узлами системы в случае использования обычных 2D камер.

Rigid body tracking for real-time system of object’s motion analyses, data obtains from one-dimension optical detectors / *Lobiv I. V.*, *Mekhontsev D. Yu.*, *Murzin F. A.* // Proc. of school-and-contest of young researchers “Theoretical and applied problems of informatics: novel approaches and solutions”. — Novosibirsk, 2003. — P. 54–62.

There are some fixed markers on a rigid body. Also there are one-dimension cameras (detectors) placing around the body. The rigid body is within detector's fields of vision. The method of detecting the rigid body is described. Data obtains from one-dimensional detectors. The sub-definiteness (lack of data) and over-definiteness (a lot of data) cases are considered. Proposed method can solve the main problem in the area. It can reduce the giant information data traffic (about 3GB per second) are obtaining from 2D cameras.

Построение распределенных объектно-ориентированных интегрированных информационных систем обеспечения безопасности предприятия / Рыжов В. С. // Тр. школы-конкурса молодых ученых ИСИ СО РАН «Теоретические и прикладные задачи информатики: новые подходы и решения». — Новосибирск, 2003. — С. 63–74.

Рассматривается модель многоуровневой информационной системы, интегрирующей разнородные классы объектов различной конфигурации и степени сложности. Автором предложено архитектурное решение, позволяющее динамически расширять действующую систему, подключая к ней объекты как уже имеющихся в ней классов, так и объекты новых классов. В работе также обобщается опыт практической реализации модели в виде распределенного приложения. Использование распределенной архитектуры позволяет значительно повысить устойчивость системы. Анализируется опыт эксплуатации разработанных систем, включающих в себя подсистемы контроля доступа, энергоснабжения, пожарной безопасности, видеонаблюдения и других систем жизнеобеспечения.

Design and implementation of distributed object-oriented integrated information systems for providing building security / Ryzhov V. S. // Proc. of school-and-contest of young researchers "Theoretical and applied problems of informatics: novel approaches and solutions". — Novosibirsk, 2003. — P. 63–74.

In the work a model of multilevel information system is considered that integrates heterogeneous classes of objects of different configuration and complexity range. The architectural solution is suggested that allows dynamical extension of working system by connecting objects both of existing classes and new classes as well. Using distributed architecture improves significantly reliability of the system. There is also considered in the work the experience of practical implementation of the model in form of distributed application and exploitation of the system that includes subsystems of access control, power supply, fire safety, video observation and other subsystems of life support

**ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ ЗАДАЧИ ИНФОРМАТИКИ:
НОВЫЕ ПОДХОДЫ И РЕШЕНИЯ**

**Труды школы-конкурса молодых ученых
Новосибирск, Академгородок
25 мая–15 июня 2003 года**

Рукопись поступила в редакцию 30.06.03
Редактор З. В. Скок

Подписано в печать 15.12.03
Формат бумаги 60 × 84 1/16
Тираж 75 экз.

Объем 4.6 уч.-изд.л., 5.0 п.л.

ЗАО РИЦ «Прайс-курьер»
630090, г. Новосибирск, пр. Акад. Лаврентьева, 6, тел. (383-2) 34-22-02