

МОДЕЛЬ УПРАВЛЕНИЯ АГЕНТАМИ В ОБЩЕЙ СЕМАНТИЧЕСКОЙ ПАМЯТИ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Зотов Н. В.

Кафедра интеллектуальных информационных технологий,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: n.zotov@bsuir.by

Предложена математическая модель управления агентами в общей семантической памяти интеллектуальных систем.

ВВЕДЕНИЕ

Платформа OSTIS [1, 2] представляет собой комплексное решение для разработки ostis-систем. Ключевыми компонентами таких систем являются интеллектуальные агенты — автономные программные компоненты, способные реагировать на события в sc-памяти и выполнять действия в соответствии со спецификациями.

Цель настоящей работы состоит в описании математической модели управления агентами в sc-памяти ostis-систем.

I. МОДЕЛЬ УПРАВЛЕНИЯ АГЕНТАМИ В SC-ПАМЯТИ

Модель управления агентами в sc-памяти определяется как:

$$\mathcal{AM} = \langle \mathcal{A}, \mathcal{S}, \mathcal{E}, \mathcal{C}, \mathcal{T}, \mathcal{R}, \mathcal{I} \rangle \quad (1)$$

где \mathcal{A} — множество агентов и их спецификаций; \mathcal{S} — пространство состояний агентов; \mathcal{E} — множество событий в sc-памяти; \mathcal{C} — модель синхронизации агентов; \mathcal{T} — временная модель агентов; \mathcal{R} — модель исполнения агентов; \mathcal{I} — программный интерфейс управления агентами.

Агенты и их спецификации образуют систему непересекающихся множеств:

$$\mathcal{A} = \bigcup_{i=1}^n \mathcal{A}_i, \text{ где } \mathcal{A}_i \cap \mathcal{A}_j = \emptyset \text{ для } i \neq j \quad (2)$$

Каждый агент $a \in \mathcal{A}$ определяется формально через его спецификацию:

$$\text{spec}(a) = \langle \tau, \pi, \alpha, \kappa, \rho, \sigma, \theta \rangle \quad (3)$$

где $\tau: \mathcal{E} \rightarrow \{0,1\}$ — предикат инициирования события в sc-памяти, определяющий первичное условие инициирования агента; $\pi: \mathcal{S} \rightarrow \{0,1\}$ — предикат активации агента, определяющий условие активации агента при успешном выполнении τ ; $\alpha: \mathcal{S} \times \mathcal{E} \rightarrow \mathcal{S}$ — функция перехода между состояниями агентов, определяющая поведение агента; $\kappa \subseteq \mathcal{E}$ — множество ключевых sc-элементов (элементов в sc-памяти), необходимых для функционирования агента; $\rho: \mathcal{S} \rightarrow \{0,1\}$ — предикат резуль-

тирующего условия агента, определяющий условие успешного выполнения агента; $\sigma \in \mathcal{P}(\mathcal{M})$ — множество объектов синхронизации.

Пространство состояний агентов задаётся:

$$\mathcal{S} = \mathcal{S}_{\text{idle}} \cup \mathcal{S}_{\text{ready}} \cup \mathcal{S}_{\text{active}} \cup \mathcal{S}_{\text{wait}} \cup \mathcal{S}_{\text{done}} \quad (4)$$

$$\forall i, j \in \{\text{idle, ready, active, wait, done}\}, i \neq j:$$

$$\mathcal{S}_i \cap \mathcal{S}_j = \emptyset \quad (5)$$

Функции перехода определяются как:

$$\begin{aligned} \text{init: } & \mathcal{S}_{\text{idle}} \times \text{spec}(\mathcal{A}) \rightarrow \mathcal{S}_{\text{ready}}, \text{ activate: } \mathcal{S}_{\text{ready}} \times \mathcal{E} \rightarrow \mathcal{S}_{\text{active}}, \\ \text{suspend: } & \mathcal{S}_{\text{active}} \rightarrow \mathcal{S}_{\text{wait}}, \text{ resume: } \mathcal{S}_{\text{wait}} \times \mathcal{E} \rightarrow \mathcal{S}_{\text{active}}, \\ \text{complete: } & \mathcal{S}_{\text{active}} \rightarrow \mathcal{S}_{\text{done}}, \text{ destroy: } \mathcal{S}_{\text{done}} \rightarrow \mathcal{S}_{\text{idle}}. \end{aligned}$$

Модель синхронизации агентов задаётся:

$$\mathcal{C} = \langle \mathcal{L}, \mathcal{Q}, \mathcal{W}, \Delta, \Gamma, \Phi \rangle \quad (6)$$

где \mathcal{L} — множество блокировок для разделяемых ресурсов; $\mathcal{Q}: \mathcal{A} \rightarrow \mathbb{N}$ — функция приоритета агентов для планирования выполнения; $\mathcal{W}: \mathcal{A} \times \mathcal{L} \rightarrow \{0,1\}$ — предикат ожидания ресурса агентом; $\Delta: \mathcal{L} \times \mathcal{A} \times \mathbb{T} \rightarrow \{0,1\}$ — предикат владения блокировкой агентом в момент времени; $\Gamma: \mathcal{A} \times \mathcal{A} \rightarrow \{0,1\}$ — отношение зависимости между агентами (граф ожидания); $\Phi: \mathcal{A} \rightarrow \mathbb{R}^+$ — функция времени ожидания агента.

Модель удовлетворяет инвариантам:

Инвариант 1 (Безопасность).

$$\forall l \in \mathcal{L}, \forall t \in \mathbb{T}: |\{a \in \mathcal{A} \mid \Delta(l, a, t) = 1\}| \leq 1 \quad (7)$$

Инвариант 2 (Живость).

$$\begin{aligned} \forall a \in \mathcal{A}, \forall t \in \mathbb{T}: & \nu(a, t) = \text{active} \implies \\ & \exists t' > t: \nu(a, t') = \text{done} \end{aligned} \quad (8)$$

Инвариант 3 (Отсутствие дедлоков).

$$\begin{aligned} \nexists (a_1, a_2, \dots, a_k) \in \mathcal{A}^k: & \Gamma(a_1, a_2) \wedge \Gamma(a_2, a_3) \wedge \\ & \wedge \dots \wedge \Gamma(a_k, a_1) = 1 \end{aligned} \quad (9)$$

Инвариант 4 (Справедливость).

$$\begin{aligned} \forall a \in \mathcal{A}, \forall l \in \mathcal{L}, \forall t \in \mathbb{T}: & \mathcal{W}(a, l) = 1 \implies \\ & \exists t' > t: \Delta(l, a, t') = 1 \end{aligned} \quad (10)$$

Временная модель выполнения задаётся:

$$\mathcal{T} = \langle \mathbb{T}, <, \nu, \eta, \mu \rangle \quad (11)$$

где \mathbb{T} – множество временных точек (дискретное или непрерывное); $< \subseteq \mathbb{T} \times \mathbb{T}$ – строгое линейное упорядочение на \mathbb{T} ; $\nu: \mathcal{A} \times \mathbb{T} \rightarrow \mathcal{S}$ – функция состояния агента во времени; $\eta: \mathcal{E} \times \mathbb{T} \rightarrow \{0,1\}$ – предикат возникновения события в момент времени; $\mu: \mathcal{A} \times \mathbb{T} \rightarrow \mathbb{R}_0^+$ – функция времени выполнения агента.

Модель исполнения агентов определяется как система переходов:

$$\mathcal{R} = \langle \mathcal{S}, \Sigma, \rightarrow, s_0, \mathcal{F} \rangle \quad (12)$$

где $\Sigma = \mathcal{E} \cup \{\tau\}$ – алфавит действий (τ обозначает внутреннее действие агента); $\rightarrow \subseteq \mathcal{S} \times \Sigma \times \mathcal{S}$ – отношение переходов, определяющее возможные переходы между состояниями; $s_0 \in \mathcal{S}_{idle}$ – начальное состояние агента; $\mathcal{F} \subseteq \mathcal{S}_{done}$ – множество допустимых финальных состояний.

Программный интерфейс управления агентами определяется множеством операций:

$$\mathcal{I} = \langle \mathcal{O}_{lifecycle}, \mathcal{O}_{control}, \mathcal{O}_{subscription} \rangle \quad (13)$$

где $\mathcal{O}_{lifecycle}$ – операции управления жизненным циклом агентов; $\mathcal{O}_{control}$ – операции управления выполнением агентов; $\mathcal{O}_{subscription}$ – операции управления подписками на события.

Множество операций жизненного цикла:

$$\mathcal{O}_{lifecycle} = \{createBySpec, deleteBySpec, initialize, start, pause, resume, stop, terminate\} \quad (14)$$

где:

$$\begin{aligned} createBySpec: Spec \rightarrow \mathcal{A} \cup \{\perp\}, & deleteBySpec: Spec \rightarrow \{\top, \perp\}, \\ initialize: \mathcal{A} \times Spec \rightarrow \{\top, \perp\}, & start: \mathcal{A} \rightarrow \{\top, \perp\}, \\ pause: \mathcal{A} \rightarrow \{\top, \perp\}, & resume: \mathcal{A} \rightarrow \{\top, \perp\}, \\ stop: \mathcal{A} \rightarrow \{\top, \perp\}, & terminate: \mathcal{A} \rightarrow \{\top, \perp\} \end{aligned}$$

где \perp обозначает неудачное выполнение операции, \top – успешное.

Множество операций управления:

$$\mathcal{O}_{control} = \{acquireLock, releaseLock, waitForEvent\} \quad (15)$$

где:

$$\begin{aligned} acquireLock: \mathcal{A} \times \mathcal{L} \rightarrow \{\top, \perp\}, \\ releaseLock: \mathcal{A} \times \mathcal{L} \rightarrow \{\top, \perp\}, \\ waitForEvent: \mathcal{A} \times \mathcal{E} \times \mathbb{R}_0^+ \rightarrow \{\top, \perp\}. \end{aligned}$$

Система подписок определяется как отношение между агентами и событиями:

$$SUB \subseteq \mathcal{A} \times \mathcal{E} \times \mathcal{FP} \times \mathcal{CF} \quad (16)$$

где \mathcal{FP} – множество предикатов фильтрации событий; \mathcal{CF} – множество функций обратного вызова для обработки событий.

Операции управления подписками:

$$\mathcal{O}_{subscription} = \{\text{subscribe}, \text{unsubscribe}\} \quad (17)$$

где:

$$\begin{aligned} \text{subscribe}: \mathcal{A} \times \mathcal{E} \times \mathcal{FP} \times \mathcal{CF} \rightarrow \text{SID} \cup \{\perp\}, \\ \text{unsubscribe}: \mathcal{A} \times \text{SID} \rightarrow \{\top, \perp\}. \end{aligned}$$

Инвариант 5 (Уникальность подписок).

$$\forall s_1, s_2 \in \text{SID}: s_1 \neq s_2 \implies \text{sub}(s_1) \neq \text{sub}(s_2) \quad (18)$$

Инвариант 6 (Консистентность подписок).

$$\begin{aligned} \forall a \in \mathcal{A}, s \in \text{SID}: \text{unsubscribe}(a, s) = \top \implies \\ \exists e, f, c: (a, e, f, c) \in SUB \end{aligned} \quad (19)$$

Инвариант 7 (Живость подписок).

$$\begin{aligned} \forall (a, e, f, c) \in SUB, \forall t \in \mathbb{T}: \eta(e, t) = 1 \wedge \\ \wedge \text{active_sub}(a, e, f, c, t) = 1 \implies \\ \exists t' \geq t: |t' - t| < \Delta_{max} \wedge c(e, t') \end{aligned} \quad (20)$$

где Δ_{max} – максимальное время доставки уведомления.

ЗАКЛЮЧЕНИЕ

Представленная модель обеспечивает формальную основу для реализации модуля управления агентами в Платформе OSTIS с гарантиями корректности через систему инвариантов, поддержку событийно-ориентированной архитектуры и полный набор операций управления жизненным циклом агентов и подписками на события.

1. Zotov, N. OSTIS Platform – a Framework for Developing Intelligent Agents Based on Semantic Networks / N. Zotov // Open Semantic Technologies for Intelligent Systems (OSTIS) : сборник научных трудов / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : В. В. Голенков [и др.]. – Минск, 2025. – Вып. 9. – С. 113-134.
2. Зотов, Н. В. Модель управления процессами в общей семантической памяти интеллектуальных систем / Н. В. Зотов // Информационные технологии и системы 2023 (ИТС 2023) : материалы Международной научной конференции, Минск, 22 ноября 2023 / Белорусский государственный университет информатики и радиоэлектроники ; редкол. : Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2023. – С. 53-54.