

Бабич М.Ю. Замкнутое программное обеспечение для многоагентных организационно-технических систем. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XVIII Междунар. научно-техн. конф. – Пенза: ПДЗ, 2018. – С. 9-16.

УДК 004.89

## ЗАМКНУТОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ МНОГОАГЕНТНЫХ ОРГАНИЗАЦИОННО-ТЕХНИЧЕСКИХ СИСТЕМ

М.Ю. Бабич

## CLOSED-LOOP SOFTWARE FOR MULTIAGENT ORGANIZATIONAL-TECHNICAL SYSTEMS

M.Yu. Babich

**Аннотация.** Рассмотрены управление и планирование работ рациональных агентов в подсистеме управления. Проанализированы ошибки, возникающие в процессе управления, а также в результате использования лицами, принимающими решение, средств вычислительной техники. Дано определение замкнутого программного обеспечения, описано его применение. Приведены принципы нового подхода, облегчающего принятие решений в процессе управления.

**Ключевые слова:** рациональный агент, многоагентные системы, организационно-технические системы, программное обеспечение, подсистема управления.

**Abstract.** The work considers the planning and control of rational agents in the control subsystem. The errors that occur during the control process and as a result of the use of computer technology by decision-makers are analyzed. The definition of closed-loop software is given, its application is described. The principles of the new approach facilitating decision making during the control process are presented.

**Keywords:** rational agent, multiagent systems, organizational-technical systems, software, control subsystem.

Рассмотрим две конкурирующие организационно-технические многоагентные системы  $S_1, S_2$ . Системы пытаются достичь цели  $P_1, P_2$ , которые одновременно недостижимы на некотором отрезке времени:  $(P_1 = \text{true} \Rightarrow P_2 = \text{false}) \wedge (P_2 = \text{true} \Rightarrow P_1 = \text{false})$ . Одной из важных подсистем каждой из рассматриваемых систем является подсистема управления (ПУ), осуществляющая мягкое управление рациональными агентами, то есть ПУ ставит перед агентами цели, достижения которых в конечном итоге должны привести к достижению цели  $P_1$  для  $S_1$  или  $P_2$  для  $S_2$ .

Информация о состоянии системы и среды поступает в ПУ от подсистемы наблюдения, в которую включены средства наблюдения, начиная от датчиков, сканирующих параметры системы, среды, и кончая докладами рациональных агентов, наблюдающих текущую обстановку.

Каким образом ПУ осуществляет управление системой, включая агентов?

Обозначим через  $S(t)$  состояние системы в момент времени  $t$ , через  $C$  – среду, в которой функционирует система  $S$ , через  $C(t)$  – состояние среды. Для системы  $S_1$  в среду входит в том числе система  $S_2$ , также как для системы  $S_2$ , в среду входит система  $S_1$ .

$$A_1 \subset S_1 \subset C_2, A_2 \subset S_2 \subset C_1, \quad (1)$$

где  $A$  – множество рациональных агентов.

Будем рассматривать процессы планирования и управления для системы  $S_1$ .

В результате воздействия ПУ на систему и среду, а также в результате других факторов изменение системы и среды происходит по траектории

$$(S_1(t_1), C_1(t_1)), (S_1(t_2), C_1(t_2)), \dots, (S_1(t_n), C_1(t_n))). \quad (2)$$

Для ПУ состояниями системы и среды являются поступающие в ПУ значения некоторых параметров, считающихся определяющими для изучения текущей обстановки, то есть траектория (2) для ПУ представляется как

$$(S_1(t_1), C_1(t_1))^m, (S_1(t_2), C_1(t_2))^m, \dots, (S_1(t_n), C_1(t_n))^m, \quad (3)$$

где  $m$  – количество выбранных параметров наблюдения. В этом случае возникает первая ошибка  $\varepsilon_1$ , так как выбранные параметры могут не совсем адекватно отражать состояние системы и среды. Ошибка  $\varepsilon_1$  существенно возрастает, когда система входит в период хаоса, так как в этом случае незначительные изменения ранее несущественных параметров могут оказать неожиданное влияние на текущую обстановку. Состояние хаоса может быть спланировано заранее. Например, для силовых структур период столкновений с противником близок к состоянию хаоса.

Кроме того, возникает ошибка  $\varepsilon_2$ , так как не вся передаваемая в ПУ информация точно отражает действительные значения выбранных параметров. Например, состояния системы  $S_2(t)$ .

На основе информации, поступившей в ПУ, предпринимается попытка определить дальнейшие действия по управлению ПУ и спрогнозировать последующие состояния системы и среды. Планирование действий и прогнозирование состояний сложной системы, которой является система  $S_1$ , зачастую пытаются осуществить, формализовав процессы, протекающие в  $S_1(t)$ ,  $C_1(t)$ , используя какой-либо современный математический аппарат для решения сформулированных задач, который реализуется на средствах вычислительной техники. Такие попытки не учитывают ошибки  $\varepsilon_1$ ,  $\varepsilon_2$ . Информация, обрабатываемая в ПУ, является неточной, неполной, нечеткой. Существенным является условие (1) и то, что под рациональным агентом в рассматриваемых системах понимается человек или человек, управляющий техническим устройством. Ответные действия ПУ системы  $S_2$  после управляющего воздействия ПУ системы  $S_1$  непредсказуемы. Вследствие этого, как правило, отсутствует статистика, на которой должна быть проверена адекватность используемых математических моделей, а список входных параметров для решения поставленных задач не является исчерпывающим. В результате либо решение принимается без учета выданной средствами вычислительной техники информации, являющейся результатом решения формализованных задач, либо прогнозируемые состояния системы и среды после управляющего воздействия со стороны ПУ значительно отличаются от действительных. Возникает ошибка  $\varepsilon_3$ .

Таким образом, управляющее воздействие со стороны ПУ приводит к состояниям  $S_1(t_i)$ ,  $C_1(t_i)$ , которые затем переходят в состояния  $S_1(t_{i+1})$ ,  $C_1(t_{i+1})$ :

$$S_1(t_i), C_1(t_i) \rightarrow S_1(t_{i+1}), C_1(t_{i+1}). \quad (4)$$

Лицо, принимающее решение (ЛПР), в ПУ предполагает, что управляющее воздействие приведет к состояниям  $(S_1^*(t_i), C_1^*(t_i))^m$ , которые перейдут в состояния  $S_1^*(t_{i+1}), C_1^*(t_{i+1})$ , то есть

$$(S_1^*(t_i), C_1^*(t_i))^m \rightarrow S_1^*(t_{i+1}), C_1^*(t_{i+1}), \quad (5)$$

$$\rho((S_1^*(t_{i+1}), C_1^*(t_{i+1})), (S_1(t_{i+1}), C_1(t_{i+1}))) + \varepsilon_1 + \varepsilon_2 + \varepsilon_3 < \varepsilon, \quad (6)$$

где  $\rho$  – мера близости;  $\varepsilon$  – планируемая незначительная ошибка. Однако условие (6) за счет ошибок  $\varepsilon_1, \varepsilon_2, \varepsilon_3$  часто не выполняется [1].

В период, близкий к состоянию хаоса с наличием большой неопределенности, решение, принимаемое ЛПР без учета результатов задач, выдаваемых средствами вычислительной техники, основанное на интуиции и опыте ЛПР, зачастую бывает более быстрым и эффективным, чем решение, полученное с использованием математических моделей и традиционной логики, от которой невозможно уйти в рамках работы вычислительных машин.

Необходимо найти другие подходы, облегчающие принятие решений в описанных ситуациях.

Одним из таких подходов является использование технологии, которая не выдает ЛПР готовые решения, а тренирует его когнитивные способности в аналогичных ситуациях, с которыми сталкивается ЛПР, и ориентирует его в пространстве принятых ранее решений.

Новый подход включает реализацию замкнутого программного обеспечения (ПО), создание на его основе игровой ситуации между виртуальными агентами из  $A_1, A_2$ , использование метода прецедентов, который обеспечивает накапливание игровых решений, и возможность их анализа в ситуациях различного типа [2].

Более подробно остановимся на технологии замкнутого программного обеспечения.

Функционирование ПО и работу должностных лиц (ДЛ) в ПУ можно описать машиной тестирования (МТ). Здесь и далее мы придерживаемся терминологии, введенной в работах [3, 4]. МТ можно представить направленным графом, состоящим из вершин, соответствующих некоторым функциям ДЛ и ПО, соединенных направленными, нагруженными дугами [5]. Так как в нашем случае будет важно время выполнения функций, то МТ будем представлять модифицированной сетью Петри (МСП) [6], переходы которой нагружены определенным для каждой функции временным периодом  $\tau$ . МСП рассматривается с некоторыми ограничениями [7]. Вершина (место) может быть активной, если началось выполнение функции, соответствующей данной вершине, или неактивной – действие закончилось. Активность переходит по направленной дуге через выходной переход к другой вершине после окончания соответствующего периода  $\tau$ .

Пусть  $L$  – алфавит символов. Сообщение, передаваемое из окружения в МТ, называется стимулом и обозначается  $?m$ , а сообщение, передаваемое из системы в окружение, называется реакцией и обозначается  $!m$ , где  $m$  – символы сообщения из алфавита  $L$ . Например, если рассматривается МТ деятельности ДЛ, то его окружением будем считать ПО, функционирующее в ПУ. Для МТ, имитирующей функции программной компоненты, окружение – деятельность ДЛ, другие программы. Обозначим через  $\omega$  символ внутреннего действия, не являющегося стимулом или реакцией, например, продолжительная обработка ПО введенных данных ( $?m$ ) перед выдачей результатов ( $!m$ ) или обсуждение ДЛ результатов работы ПО. Определим биекцию  $b$  и переменную  $w$ :  $(b(?m) = !m) \wedge (b(!m) = ?m)$ ,  $(w=?m) \vee (w=!m)$ . Каждый переход в МТ (МСП) дополнительно помечен  $w$  или  $\omega$ , то есть по окончании активности вершины выходной переход имитирует выдачу какой-либо реакции, или считывание какого-либо стимула, или завершение внутреннего действия.

Рассматривая многоагентную систему вместе с ее подсистемами и другими взаимодействующими системами, устройствами, мы получим объединение МТ и алфавитов:

$$N = \left( \bigcup_{i=1}^l N_i \right) \cup N_{cp}; \quad L = \left( \bigcup_{i=1}^l L_i \right) \cup L_{cp}, \quad (7)$$

где  $N_i$  – это МТ, соответствующие функционированию программ, подсистем, устройств, ДЛ, взаимодействующих систем (всего  $l$ );  $N_{cp}$  – МТ среды;  $L_i, L_{cp}$  – алфавиты, соответствующие МТ  $N_i$  и  $N_{cp}$ . МТ  $N$  имитирует функционирование системы  $S$  и среды  $C$ . Операция объединения  $\cup$  в (7) и далее для МТ понимается как операция наложения МСП [8], когда нет общих вершин, дуг и переходов.

Переход, помеченный  $w$  или  $\omega$ , будем называть асинхронным относительно машин тестирования  $N_i, N_j$ , если  $w \in (L_i \setminus b(L_j)) \cup (L_j \setminus b(L_i))$ .

Переход будем называть абсолютно асинхронным, если для любых  $i \neq j$  он является асинхронным переходом относительно МТ  $N_i, N_j$ .

Переходы, помеченные  $w_1, w_2$ , принадлежащие МТ  $N_i, N_j$ , будем называть квазисинхронными относительно машин тестирования  $N_i, N_j$ , если

$$w_1, w_2 \in (L_i \cap b(L_j)) \cup (L_j \cap b(L_i)). \quad (8)$$

Квазисинхронные переходы, помеченные как  $w_1, w_2$ , причем  $w_2 = b(w_1)$ , будем называть синхронными относительно времени  $t$ , если

$$|t_{w_1} + \tau_{1w} - (t_{w_2} + \tau_{2w})| \leq t, \quad (9)$$

где  $t_{w_1}, t_{w_2}$  – время начала активности вершин, выходные переходы которых помечены  $w_1$  и  $w_2$  в МТ  $N_i$  и  $N_j$ ;  $\tau_{1w}, \tau_{2w}$  – время срабатывания переходов, для которых эти вершины являются входными;  $t$  – заданное время.

Рассмотрим такое ПО, функционирование которого имитируется МТ  $N$ , где  $N$  определяется (7), что в МТ  $N$  из абсолютно асинхронных переходов присутствуют только абсолютно асинхронные переходы, помеченные  $\omega$ .

Утверждение. Все переходы, не помеченные  $\omega$ , являются квазисинхронными относительно некоторых машин тестирования  $N_i, N_j$ , где  $(i \leq l+1) \wedge (j \leq l+1)$ .

Если для каждого перехода из каждой пары квазисинхронных переходов можно найти синхронный переход, то такое ПО будем называть замкнутым.

Наша цель изменить ПО таким образом, чтобы внешней средой, раз в ней присутствуют рациональные агенты, управляли ДЛ из группы подыгрыша. Назовем их тестерами. В качестве имитационной модели будем рассматривать само ПО, установленное на ПУ и работающее в реальном режиме, а также действия ДЛ, в соответствии с их инструкциями, уставами, правилами и т.д. ПО должно быть несколько модифицировано, но таким образом, чтобы модификация по сравнению с рабочей версией была минимальной. Для этого заметим, что ПО ПУ воспринимает среду через систему протоколов взаимодействия с устройствами, ее сканирующими. Исключим из ПО все программные компоненты, непосредственно обрабатывающие поступившую информацию в данные, «понятные» устройствам сканирования, и дополним программными компонентами, позволяющими тестерам изменять в процессе работы ПО значения переменных протоколов взаимодействия и тем самым управлять средой  $C$ . Предположим, что количество МТ

действий тестеров равно количеству МТ, имитирующих работу программных компонент.

Вместо (7) рассмотрим

$$N = \left( \bigcup_{i=1}^k N_i \right) \cup \left( \bigcup_{i=k+1}^g N_i^H \right),$$

$$N^T = \left( \bigcup_{i=1}^k N_i^T \right) \cup \left( \bigcup_{i=k+1}^g N_i^{TH} \right), \quad 2g = 1, \quad (10)$$

$$L = \left( \bigcup_{i=1}^k L_i \right) \cup \left( \bigcup_{i=k+1}^g L_i^H \right), \quad L^T = \left( \bigcup_{i=1}^k L_i^T \right) \cup \left( \bigcup_{i=k+1}^g L_i^{TH} \right),$$

$$R = N \cup N^T \cup N^{cp}, \quad (11)$$

где  $N_i$  – МТ, имитирующие работу оставшегося ПО после удаления  $g - k$  программных компонент, обрабатывающих информацию в данные, «понятные» устройствам сканирования;  $N_i^H$  – МТ, имитирующие работу новых программных компонентов, управляющих средой;  $N_i^T$ ,  $N_i^{TH}$  – МТ действий тестеров, работающих с программными компонентами, функционирование которых имитируют МТ  $N_i$  и  $N_i^H$ ;  $L$ ,  $L^T$  – соответствующие алфавиты;  $N^{cp}$  – МТ, имитирующая функционирование оставшейся части среды, находящейся вне модели  $N \cup N^T$ , для которой отсутствуют компоненты, имитируемые МТ  $N_i^H$ .

Идея применения замкнутого ПО заключается в следующих требованиях:

а) ПО должно быть замкнутым для МТ  $R$ ,

б)  $\rho(C_M^*, C^*) < \varepsilon$ ,

где  $C^*$  – множество последовательностей состояний среды;  $C_M^*$  – множество последовательностей состояний среды, которые возможно моделировать программными компонентами, чью работу имитируют МТ  $N_i^H$ .

Например, было создано ПО для системы слежения за беспилотными летающими аппаратами (БПЛА). Оно было дополнено программными компонентами, моделирующими движение БПЛА на электронной карте местности. Координаты виртуальной БПЛА отсылались в ПО по реальным протоколам взаимодействия. Движением виртуальным БПЛА управлял тестер со своего компьютера. ПО было спроектировано замкнутым, и все возможные траектории реального БПЛА могли совпадать с траекториями виртуального БПЛА, которым управлял тестер.

Использование замкнутого ПО позволяет не только применять новую технологию в процессе принятия решений, но и упрощает процесс комплексного тестирования и обучения должностных лиц, работающих с ПО.

#### Библиографический список

1. Бабич М.Ю. К вопросу интеллектуализации специальных организационно-технических систем // Вопросы радиоэлектроники. 2017. № 12. С. 6 – 12.
2. Бабич М.Ю. Нерешаемые задачи интеллектуальной поддержки управления многоагентными организационно-техническими системами // XXI век: итоги прошлого и проблемы настоящего плюс. Серия: «Технические науки. Информатика, вычислительная техника и управление». 2017. Вып. № 4 (38). С. 71 – 78.
3. R. Milner. Modal characterisation of observable machine behaviour. In G. Astesiano & C. Bohm, editors: Proceedings CAAP 81, LNCS 112, Springer, pp. 25 – 34.

4. Бурдонов И.Б., Косачев А.С., Кулямин В.В. Теория соответствия для систем с блокировками и разрушением. М.: Физматлит, 2008. 494 с.

5. Кузьмин Е.В., Соколов В.А. Структурированные системы переходов. М.: Физматлит, 2006. 176 с.

6. Криницкий Н.А., Федотова Д.Э., Квасова Т.К. О едином алгоритмическом подходе к аппаратным и программным средствам вычислительных систем и их моделировании // Программирование. 1985. № 5. С. 12–19.

7. Бабич М.Ю. Вопросы взаимодействия моделей автоматизированных информационных систем управления // XXI век: итоги прошлого и проблемы настоящего плюс. Серия: «Технические науки». 2012. Выпуск №5 (09). С. 97 – 102.

8. Котов В.Е. Сети Петри. М.: Наука, 1987. 160 с.

**Бабич Михаил Юрьевич**

АО «НПП «Рубин»,

г. Пенза, Россия

E-mail: babichmj@mail.ru

**Babich M.Yu.**

SC «SIE «Rubin»,

Penza, Russia