# АГЕНТНОЕ МОДЕЛИРОВАНИЕ В ЗАДАЧЕ АВТОМАТИЗИРОВАННОЙ РАЗРАБОТКИ ПЛАНОВ ЛОКАЛИЗАЦИИ АВАРИЙ ОПАСНЫХ ПРОИЗВОДСТВЕННЫХ ОБЪЕКТОВ МЕТАЛЛУРГИЧЕСКИХ ПРЕДПРИЯТИЙ

**Б. В. Панов (Череповец)**

Согласно требованиям нормативных документов для каждого газоопасного и взрывопожароопасного производства, цеха, отделения, участка, технического устройства металлургического предприятия должны быть разработаны планы локализации аварий (ПЛА), которые определяют действия производственного персонала, сервисных служб и аварийно-спасательных формирований по локализации возможных аварий на отдельных стадиях их развития.

Разработка ПЛА является сложным, трудно формализуемым и, как следствие, слабо автоматизированным процессом, требующим анализа большого объема нормативной и технической документации, а также информации, не имеющей документального оформления – знаний специалистов. Рассматриваемый метод автоматизированной разработки ПЛА позволяет автоматизировать и тем самым значительно упростить и упорядочить процесс разработки ПЛА путем введения унифицированных автоматизированных этапов разработки: 1) разработка онтологии; 2) генерация ПЛА; 3) оценка качества ПЛА.

Первый этап предполагает создание онтологического описания предметной области на основе продукционно-фреймового представления знаний. Введено понятие фрейм-концепта (ФК) [1], который позволяет инкапсулировать статический и динамический аспекты описания концептов онтологии, а также связей между ними, и задается в виде четверки:

$$FK = \langle NM, SA, SL, BM \rangle, \qquad (1)$$

где $NM$ – имя ФК (символический идентификатор), $SA$ – множество атрибутов ФК, $SL$ – множество отношений ФК, $BM$ – модель поведения ФК.

Каждый ФК имеет индивидуальный набор атрибутов. Отношения ФК задаются в виде бинарных связей следующих допустимых видов: композиция, агрегация, зависимость, ассоциация. Кроме того, выделены пассивные ФК (ПФК), для которых $BM = \varnothing$, и активные ФК (АФК), для которых $BM \neq \varnothing$. При этом, ПФК рассматривается как инфообъект, простейший вид которого $\langle NM, \varnothing, \varnothing, \varnothing \rangle$, задающий некий термин предметной области. АФК обладает собственным поведением и рассматривается как агент-исполнитель (далее просто агент).

В роли агентов выступают лица, обеспечивающие выполнение мероприятий, предписываемых ПЛА, а в качестве инфообъектов (среды агентов) – их окружение (производственные помещения, оборудование, средства защиты и пр.). При этом, среда функционирования агентов является замкнутой (может быть исчерпывающе описана конечным множеством инфообъектов), полностью наблюдаемой (состояние любого инфообъекта полностью доступно для агентов) и управляемой (любой инфообъект может быть переведен агентами в требуемое состояние).

Определение модели поведения агентов требует формального языка программирования агентов. В большинстве случаев в качестве такого языка используется один из универсальных языков программирования, таких, как C++ или Java. Их недостатком является относительная сложность разработки, требующая привлечения высококвалифицированных программистов. Специализированные языки программирования агентов, как правило, неполностью понятны с семантической точки зрения [2] и не в полной мере соответствуют решаемой задаче.

Предлагается оригинальный подход к программированию агентов в рамках задачи автоматизированной разработки планов локализации аварий опасных производственных объектов металлургических предприятий, отличающийся наглядностью для разработчика, четкой семантикой и простотой реализации.

Модель поведения агента задается в виде:

$$BM = \langle S, E, C, R \rangle, \quad (2)$$

где $S$ – множество состояний агента, $E$ – множество событий, воспринимаемых агентом, $C$ – множество условий реакций агента на события, $R$ – множество реакций агента на события.

Введение механизма дискретных состояний в рамках рассматриваемой архитектуры агента необходимо для синхронизации отдельных агентов, а также реализации сложного их поведения. События, поступающие на вход агента (акт восприятия) инициируют процесс принятия решения на основе внутренних убеждений (знаний) агента. При этом, двойка ($C$, $R$) из (2) рассматривается как продукция, где $C$ – посылки (условие продукции), $R$ – заключение (действие). Цель любого агента – выполнить необходимые мероприятия в рамках своих знаний и полномочий по локализации конкретной возможной аварии во взаимодействии с другими агентами.

Таким образом, онтология рассматривается как конечное множество ФК, включающее в себя ПФК (инфообъекты) и АФК (агенты):

$$Ont = \{FK_i\} = FK_P \cup FK_A,\ i=1..N, \quad (3)$$

где $FK_i$ – $i$-й ФК, $N$ – число ФК в онтологии, $FK_P$ – множество ПФК, $FK_A$ – множество АФК, $FK_P \cap FK_A = \varnothing$.

Преимущество такого описания состоит в том, что разработчику необязательно знать поведение всей системы в целом, достаточно задать поведение каждого агента. Кроме того, агентная модель органично проецируется на организационную структуру предприятия – каждый агент соответствует определенному должностному лицу.

В качестве примера рассмотрим простой фрагмент технологической схемы, представленный на рис. 1.

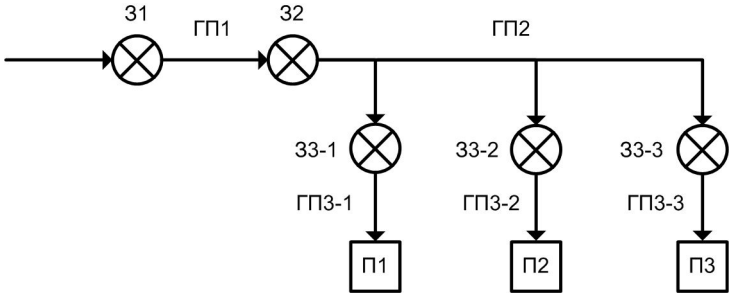


**Рис. 1. Фрагмент технологической схемы**

На схеме линиями обозначены участки газопровода, стрелками – направление движение газа. Отдельные участки газопровода соединены посредством задвижек. Кроме того, указаны потребители газа (например, нагревательные печи). Для каждого участка газопровода возможна авария вида «разрыв газопровода».

Определим состав агентов: первый заметивший, диспетчер, исполнитель. Опишем механизм их взаимодействия. Первый заметивший должен сообщить об аварии диспетчеру, который дает указание исполнителю локализовать аварию. Исполнитель, прежде всего, должен произвести отключение всех потребителей, газ к которым поставляется через аварийный участок. Далее исполнитель должен закрыть ближайшую

входную задвижку, чтобы перекрыть поступление газа в аварийный участок газопровода, используя газозащитный аппарат. После этого авария считается локализованной.

В ликвидации реальной аварии, как правило, участвует большее число исполнителей (агентов), а действия, исполняемые ими, сложнее, однако приведенный пример достаточно прост и нагляден.

Формализованное описание поведения агентов зададим в виде моделей поведения в соответствии с (2). Для рассматриваемого примера модели поведения агентов представлены на рис. 2.

**Первый заметивший**

| Состояние | Событие | Условие | Действие |
|---|---|---|---|
| S0 | Авария(Вид_аварии,Объект) | | ВЫВОД("Сообщить об аварии диспетчеру") |
| | | | Диспетчер→Сообщение_об_аварии(Вид_аварии,Объект) |

**Диспетчер**

| Состояние | Событие | Условие | Действие |
|---|---|---|---|
| S0 | Сообщение_об_аварии(Вид_аварии,Объект) | Вид_аварии=Разрыв_газопровода | ВЫВОД("Сообщить об аварии исполнителю") |
| | | | Исполнитель→Локализовать_разрыв(Объект) |
| | | ИНАЧЕ | ОШИБКА |

**Исполнитель**

| Состояние | Событие | Условие | Действие |
|---|---|---|---|
| S0 | Локализовать_разрыв(Объект) | | Дочерние_объекты=Объект:ПОИСК(∞) |
| | | | Дочерние_потребители=Дочерние_объекты:ФИЛЬТР(Потребитель) |
| | | Дочерние_потребители=∅ | ОШИБКА |
| | | | i=1 |
| | | | →Отключить_потребителя(Объект) |
| | | | СОСТОЯНИЕ(S1) |
| S1 | Отключить_потребителя(Объект) | | ВЫВОД("Отключить ",Дочерние_потребители[i].имя) |
| | | i<Дочерние_потребители:РАЗМЕР | i=i+1 |
| | | | →Отключить_потребителя(Объект) |
| | | | СОСТОЯНИЕ(S1) |
| | | | →Отключить_аварийный_участок(Объект) |
| | | | СОСТОЯНИЕ(S2) |
| S2 | Отключить_аварийный_участок(Объект) | | Входные_объекты=Объект:ПОИСК(-1) |
| | | | Входная_задвижка=Входные_объекты:ФИЛЬТР(Задвижка) |
| | | Входная_задвижка=∅ | ОШИБКА |
| | | | ВЫВОД("Закрыть ",Входная_задвижка.имя;"Газозащитный аппарат") |
| | | | КОНЕЦ |

**Рис. 2. Модели поведения агентов**

На втором этапе осуществляется автоматическая генерация ПЛА. При этом проводится имитационный эксперимент на мультиагентной модели, заданной в виде (3).

Формально ПЛА определяется как множество возможных аварийных ситуаций, для каждой из которых указывается упорядоченное множество действий, подлежащих исполнению ответственными лицами в случае наступления этой аварийной ситуации:

$$PLA = \{Sit_i\},\ Sit_i=\{Act_{i,j}\},\ i=1..N,\ j=1..H_i, \qquad (4)$$

где $Sit_i$ – $i$-я аварийная ситуация, $Act_{i,j}$ – $j$-е действие для $i$-й аварийной ситуации, $N$ – число аварийных ситуаций, $H_i$ – число действий для $i$-й аварийной ситуации.

При этом считается, что действие, имеющее больший порядковый номер, должно быть выполнено не ранее (после или одновременно) действия, имеющего меньший порядковый номер. Конкретный временной интервал между действиями, как правило, не задается. В то же время, агенты в мультиагентной среде представимы в виде параллельно работающих вычислителей. Следовательно, необходим механизм обеспечения синхронизации агентов, который позволил бы преобразовать набор асинхронных параллельных во времени актов взаимодействия агентов в упорядоченную последовательность синхронных действий в виде (4). Представленный на рис. 3 алгоритм позволяет это сделать.

При описании алгоритма использованы следующие обозначения: $S_i$ – состояние $i$-го агента; $Q_i$ – очередь сообщений $i$-го агента; $q_i$ – первоочередное сообщение из очереди сообщений $i$-го агента $Q_i$; $E^{<i>}_{S_i,j}$ – $j$-е событие для $S_i$-го состояния $i$-го агента; $C^{<i>}_{S_i,j,k}$

– условие $k$-й продукции для $j$-го события $S_i$-го состояния $i$-го агента; $R^{<i>}_{S_i,j,k}$ – действие $k$-й продукции для $j$-го события $S_i$-го состояния $i$-го агента; $N$ – число агентов; $M^{<i>}_{S_i}$ – число событий для $S_i$-го состояния $i$-го агента; $Z^{<i>}_{S_i,j}$ – число продукций для $j$-го события $S_i$-го состояния $i$-го агента; $i$, $j$, $k$ – счетчики циклов.

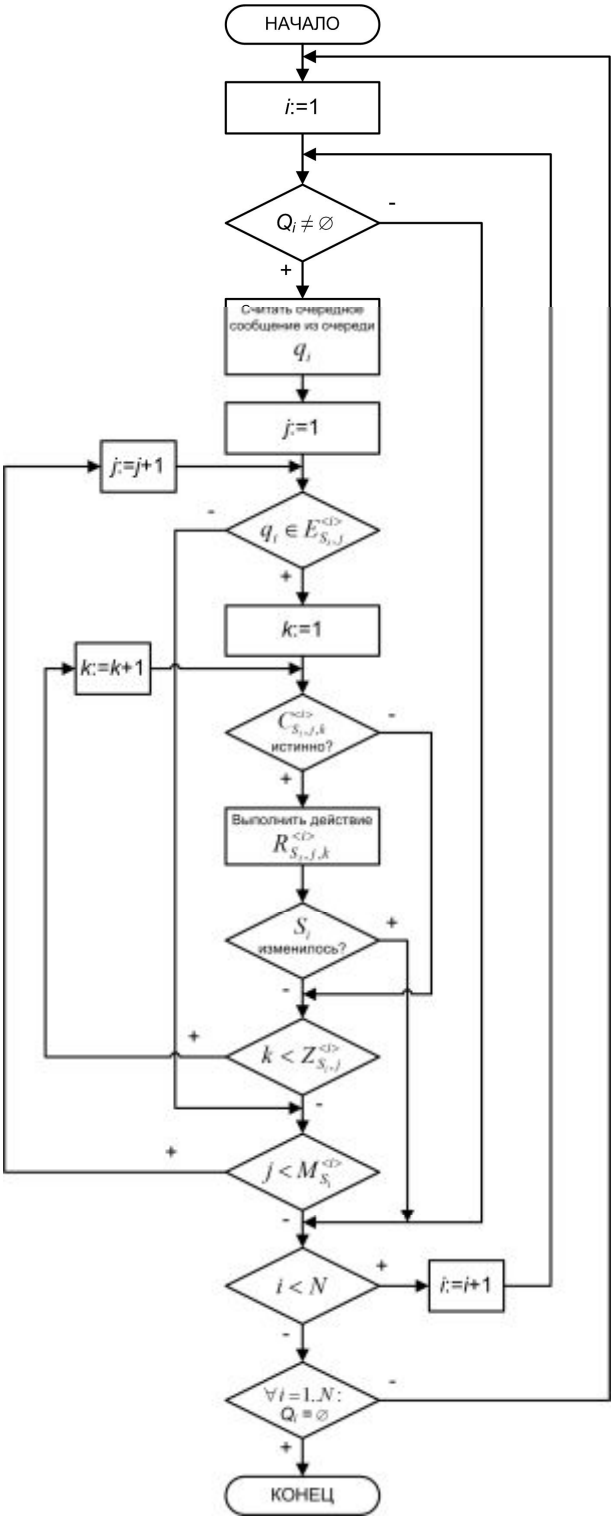


**Рис. 3. Схема алгоритма автоматической генерации ПЛА**

Начальными условиями выполнения алгоритма являются:

$$\forall\, i = 1..N:\ S_i = 0,\ Q_i = \varnothing;\ \forall\, j = 1..O\ \forall\, k = 1..T_j:\ A_{j,k} = A^0_{j,k}, \qquad (5)$$

где $A_{j,k}$ – значение $k$-го атрибута $j$-го ФК (инфообъекта или агента), $A^0_{j,k}$ – начальное значение $k$-го атрибута $j$-го ФК, $O$ – число ФК, $T_j$ – число атрибутов $j$-го ФК.

Алгоритм основан на дискретно-событийном подходе к моделированию. Все входящие для агента сообщения помещаются в очередь сообщений, организованную по принципу FIFO. Все сообщения, находящиеся в очередях сообщений агентов в рамках одного цикла исполнения (внешний цикл алгоритма), считаются одновременными во времени. Все действия, выполняемые агентами, считаются совершаемыми мгновенно. Любой агент может посылать сообщения, в том числе, и самому себе. Продукции исполняются в порядке их указания. В случае, если результатом исполнения продукции является изменение состояния агента, то управление передается следующему агенту. Таким образом, данный алгоритм позволяет реализовать все базовые типы алгоритмических структур: следование, альтернатива и цикл.

Алгоритм исполняется для каждого определенного онтологией аварийного события для каждого ассоциированного с ним объекта. При этом исходное событие помещается в очередь сообщений ассоциированного с ним агента. Как правило, этим агентом является универсальный агент «Первый заметивший», который, обладая собственным поведением, заданным моделью поведения, посылает сообщения другим агентам и тем самым запускает всю дальнейшую цепочку взаимодействия. В свою очередь, остальные агенты, обладая собственным поведением, общаются с другими агентами и/или выполняют некие манипуляции с объектами. Условием окончания исполнения алгоритма является отсутствие сообщений в очередях сообщений агентов.

Результат работы алгоритма для описываемого примера представлен в табл. 1.

**Таблица 1**

**Результат генерации ПЛА**

| **Авария** | **Последовательность мероприятий** | **Исполнители** | **Средства противоаварийной защиты** |
|---|---|---|---|
| Разрыв газопровода *ГПЗ-1* | 1. Сообщить об аварии диспетчеру | *Первый заметивший* | |
| | 2. Сообщить об аварии исполнителю | *Диспетчер* | |
| | 3. Отключить *потребителя П1* | *Исполнитель* | |
| | 4. Закрыть *задвижку ЗЗ-1* | *Исполнитель* | Газозащитный аппарат |
| Разрыв газопровода *ГПЗ-2* | 1. Сообщить об аварии диспетчеру | *Первый заметивший* | |
| | 2. Сообщить об аварии исполнителю | *Диспетчер* | |
| | 3. Отключить *потребителя П2* | *Исполнитель* | |
| | 4. Закрыть *задвижку ЗЗ-2* | *Исполнитель* | Газозащитный аппарат |
| Разрыв газопровода *ГПЗ-3* | 1. Сообщить об аварии диспетчеру | *Первый заметивший* | |
| | 2. Сообщить об аварии исполнителю | *Диспетчер* | |
| | 3. Отключить *потребителя П3* | *Исполнитель* | |
| | 4. Закрыть *задвижку ЗЗ-3* | *Исполнитель* | Газозащитный аппарат |
| Разрыв | 1. Сообщить об аварии диспетчеру | *Первый заметив-* | |

| Авария | Последовательность мероприятий | Исполнители | Средства противоаварийной защиты |
|---|---|---|---|
| газопровода *ГП2* | | *ший* | |
| | 2. Сообщить об аварии исполнителю | *Диспетчер* | |
| | 3. Отключить *потребителя П1* | *Исполнитель* | |
| | 4. Отключить *потребителя П2* | *Исполнитель* | |
| | 5. Отключить *потребителя П3* | *Исполнитель* | |
| | 6. Закрыть *задвижку З2* | *Исполнитель* | Газозащитный аппарат |
| Разрыв газопровода *ГП1* | 1. Сообщить об аварии диспетчеру | *Первый заметивший* | |
| | 2. Сообщить об аварии исполнителю | *Диспетчер* | |
| | 3. Отключить *потребителя П1* | *Исполнитель* | |
| | 4. Отключить *потребителя П2* | *Исполнитель* | |
| | 5. Отключить *потребителя П3* | *Исполнитель* | |
| | 6. Закрыть *задвижку З1* | *Исполнитель* | Газозащитный аппарат |

Необходимо отметить, что имитационный эксперимент, осуществляемый по описанному алгоритму, может завершиться неудачей. Например, агент может в бесконечном цикле посылать сообщение самому себе или другому агенту, либо обратиться к несуществующему объекту и пр. Однако подобные случаи легко отследить и выдать соответствующее предупреждение разработчику для принятия мер по устранению.

Третий этап автоматизированной разработки ПЛА предполагает оценку качества сгенерированного ПЛА, которая осуществляется в три этапа: 1) формальная оценка; 2) оценка по существу; 3) практическая оценка. Формальная оценка проводится автоматически на соответствие следующим критериям: 1) в процессе генерации ПЛА не произошло ошибок; 2) сгенерированный ПЛА включает в себя конечное число мероприятий (действий); 3) сгенерированный ПЛА завершается меткой конца. Оценка по существу проводится экспертами с использованием метода экспертной оценки. Последний этап контроля – практическая оценка – оценка качества ПЛА по результатам проведения учебных тревог.

Рассмотренный метод автоматизированной разработки ПЛА положен в основу автоматизированной информационно-обучающей системы «Электронный ПЛА», внедряемой на ЧерМК ОАО «Северсталь».

**Вывод**

Агентное моделирование позволяет решить актуальную, сложную и трудно формализуемую задачу автоматизированной разработки планов локализации аварий опасных производственных объектов металлургических предприятий. Предложена оригинальная модель онтологического описания на основе продукционно-фреймового представления знаний, подход к программированию агентов и алгоритм имитационного эксперимента, основанный на дискретно-событийном подходе к моделированию. Полученные результаты имеют практическую направленность и могут быть использованы при разработке соответствующих автоматизированных информационных систем.

**Литература**

1. **Швецов А. Н., Яковлев С. А.** Распределенные интеллектуальные информационные системы. СПб.: Изд-во СПбГЭТУ ЛЭТИ, 2003. 318 с.

2. **Черемисинов Д., Черемисинова Л.** Подход к программированию агентов в мультиагентных системах // International Book Series “Information Science and Computing”. № 4. 2008. С. 141–147.