НАУЧНОЕ ИЗДАНИЕ МГТУ ИМ. Н. Э. БАУМАНА
НАУКА и ОБРАЗОВАНИЕ
Эл № ФС77 - 48211. Государственная регистрация №0421200025. ISSN 1994-0408
электронный научно-технический журнал

Оценка надежности резервированных многоагентных систем

 # 01, январь 2014
DOI: 10.7463/0114.0696290
Игумнов А. В., Сараджишвили С. Э.
УДК 004.052.3
Россия, Санкт-Петербургский государственный политехнический университет
Alexei.Igumnov@gmail.com
Ssaradg@yandex.ru

Введение

Многоагентные технологии [1] как метод теории искусственного интеллекта предоставляют ряд преимуществ при разработке различных промышленных систем, в том числе систем управления. Одной из ключевых характеристик промышленных систем является надежность, при этом возможна постановка как задачи оценки показателей надежности существующей системы, так и задачи синтеза системы с заданными показателями надежности. Надежность технической системы, как правило, определяется ее устойчивостью к отказам, а основным средством обеспечения отказоустойчивости является резервирование [2], то есть введение избыточности. Существующие методики обеспечения отказоустойчивости многоагентных систем (МАС) Brokered MAS [3], DARX [4], Meta-Agent [5] и работа [6] основаны на избыточности агентов или избыточности агентов и некоторых задач. Такие методики подтверждают увеличение уровня надежности многоагентных систем, то есть снижение количества отказов, только экспериментально. Основным недостатком известных методик разработки отказоустойчивых МАС является отсутствие метода оценки показателей надежности МАС, разработанных с их применением, следовательно использование существующих методик не гарантирует достижения определенного уровня надежности МАС. При этом следует отметить, что в области технических систем существуют методы расчеты показателей надежности, например, логико-вероятностные методы [7]. Кроме того, существующие методики используют модели, основанные на формальном определении МАС как системы, состоящей из взаимодействующих агентов [8], хотя современные промышленные МАС являются, как правило, сложными комплексами, состоящими из аппаратных компонентов, программных агентов и исполнительных механизмов или ресурсов. Необходимость оценки показателей надежности распределенных

программно-аппаратных многоагентных систем при использовании агентного подхода к разработке промышленных систем определяет актуальность проведенного авторами исследования.

Цель работы – обеспечить оценку надежности распределенных программно-аппаратных комплексов, реализованных на основе агентно-ориентированного подхода. Научная новизна работы заключается в следующем:

- разработана модель резервированной распределенной программно-аппаратной МАС, заданная множествами задач, являющихся минимальными функциональными элементами системы, исполнительных механизмов, необходимых для выполнения задач, агентов, являющихся программными исполнительными контейнерами для задач системы, и агентных платформ, предназначенных для исполнения программных агентов;
- предложенная модель резервированной МАС учитывает не только необходимость использования задачами исполнительных механизмов, но и возможность объединения исполнительных механизмов в конвейеры, при котором отказ одного компонента конвейера приводит к отказу всего конвейера;
- разработанная модель резервированной МАС задает стратегию обеспечения отказоустойчивости, основанную на:
    - использовании резервирования задач исходной МАС, как ее минимальных функциональных элементов;
    - использовании резервирования исполнительных механизмов исходной МАС, как ресурсов, необходимых для выполнения ее задач;
    - введении избыточных множеств агентов, по которым должны быть распределены задачи резервированной МАС, и агентных платформ, по которым должны быть распределены агенты резервированной МАС, в силу нецелесообразности резервирования данных компонентов исходной МАС, так как они являются исполнительными контейнерами и, следовательно, не предназначены для реализации предопределенных функций системы;
- разработаны критерии работоспособности резервированной МАС, заданной предложенной моделью, которые необходимы для оценки надежности системы с помощью логико-вероятностных методов;
- разработана методика оценки надежности распределенных программно-аппаратных МАС с помощью логико-вероятностных методов, определяющая процедуру получения функции работоспособности системы на основе предложенных критериев работоспособности, работоспособность которой подтверждена проведенными вычислительными экспериментами, основанными на методе статистического моделирования.

Практическая ценность работы заключается в том, что разработанные модель резервированной МАС и методика оценки ее надежности могут быть использованы как для

расчета надежности технических систем, реализованных на основе многоагентного подхода, на стадии проектирования, так и в задачах синтеза оптимальной структуры распределенной программно-аппаратной МАС с заданной или максимальной надежностью.

Первые раздел статьи посвящен постановке задачи проведенного исследования. Во втором разделе описана формальная модель исходной МАС, отказоустойчивость которой необходимо обеспечить. В третьем разделе статьи определена формальная модель резервированной распределенной программно-аппаратной МАС. В четвертом разделе сформулированы критерии работоспособности резервированной МАС. Пятый раздел статьи посвящен процедуре получения логической функции работоспособности резервированной МАС, лежащей в основе методики оценки надежности МАС. В шестом разделе приведено описание проведенного вычислительного эксперимента и его основные результаты. В заключении подведены основные итоги работы, определены области применения ее результатов, а также описано дальнейшее направлении исследований в сфере обеспечения отказоустойчивости распределенных программно-аппаратных МАС.

1. Постановка задачи

Так как существующие методики обеспечения отказоустойчивости МАС ориентированы на использование только агентной модели, то есть рассматривают МАС как систему взаимодействующих агентов [8], то необходимо разработать модель программно-аппаратной МАС, учитывающую необходимость использования соответствующих аппаратных компонентов для исполнения агентной модели. Согласно [2] обеспечение отказоустойчивости технической системы должно быть основано на избыточности, следовательно необходимо разработать модель резервированной программно-аппаратной МАС, основанную на введении избыточности ее компонентов и использовании резервирования компонентов исходной МАС. Применение логико-вероятностных методов для получения оценки надежности требует разработки критериев работоспособности распределенных программно-аппаратных МАС, кроме того, необходимо определить методику получения функции работоспособности на основе разработанных критериев. В соответствии с поставленной целью работы для обеспечения оценки надежности распределенных программно-аппаратных комплексов, реализованных на основе многоагентного подхода, были сформулированы следующие задачи:

- разработать модель распределенной программно-аппаратной МАС, учитывающую необходимость использования аппаратных компонентов как для выполнения соответствующей агентной модели, так и для выполнения агентами своих функций;
- разработать стратегию обеспечения отказоустойчивости и модель резервированной распределенной программно-аппаратной МАС, основанные на использовании избыточности всех компонентов программно-аппаратной МАС;

- сформулировать критерии работоспособности резервированной распределенной программно-аппаратной МАС, необходимые для использования логико-вероятностных методов;
- разработать методику оценки надежности распределенных программно-аппаратных МАС, определяющую процедуру получения логической функции работоспособности системы на основе критериев ее работоспособности;
- экспериментально доказать работоспособность и достоверность предложенной методики оценки надежности распределенных программно-аппаратных МАС.

2. Формальная модель многоагентной системы

Многоагентная система – это система, состоящая из взаимодействующих агентов, выполняющих свои функции [8]. Разработанная модель распределенной программно-аппаратной МАС основана на модели, предложенной в работе [6], которая вводит задачу агента МАС как минимальный функциональный элемент системы и учитывает необходимость использования определенных ресурсов для выполнения агентами своих задач.

Будем считать, что агент МАС реализован программно, следовательно, МАС должна содержать аппаратные компоненты, предназначенные для исполнения программных агентов. Назовем такой аппаратный компонент агентной платформой и введем множество $HWP = \{hwp_i\}$ агентных платформ МАС. Введение множества агентных платформ в дополнение к множествам агентов $A = \{a_i\}$ и задач $T = \{tj\}$ системы позволяет задать конфигурацию программно-аппаратной МАС как распределение задач системы по ее агентам и распределение агентов по агентным платформам. Пусть распределение задач системы по агентам задано предикатом:

$$confTaskAgent : T \times A \to \{true, false\}, \quad (1)$$

где T – множество задач, A – множество агентов.

Пусть распределение агентов по агентным платформам задано предикатом:

$$confAgentHwp : A \times HWP \to \{true, false\}, \quad (2)$$

где A – множество агентов, HWP – множество агентных платформ.

Будем считать, что исполнительные механизмы программно-аппаратной МАС реализованы в виде аппаратных компонентов и являются ресурсами, необходимыми для выполнения отдельных задач системы, и введем множество $HWR = \{hwr_i\}$ исполнительных механизмов или ресурсов МАС. Введем предикат, определяющий необходимость использования определенного исполнительного механизма отдельной задачей системы:

$$reqHwrOfTask : HWR \times T \to \{true, false\}, \quad (3)$$

где HWR – множество исполнительных механизмов системы, T – множество задач МАС.

Пусть наличие связи между агентной платформой и исполнительным механизмом определяет доступность определенного исполнительного механизма для агентной платформы и,

следовательно, для всех размещенных в данной агентной платформе агентов и их задач. Введем предикат, определяющий наличие связи между исполнительным механизмом и агентной платформой:

$$confHwrHwp : HWR \times HWP \rightarrow \{true, false\}, \quad (4)$$

где *HWR* – множество исполнительных механизмов системы, *HWP* – множество агентных платформ МАС.

Разработанная формальная модель представляет программно-аппаратную МАС как кортеж *MAS* = <*T, A, HWP, HWR*>, где *T* – множество задач МАС, *A* – множество агентов, *HWP* – множество агентных платформ, *HWR* – множество исполнительных механизмов системы. Конфигурация программно-аппаратной МАС задана предикатами (1), (2) и (4), а необходимость использования исполнительных механизмов для выполнения задач задана предикатом (3).

3. Формальная модель резервированной многоагентной системы

Так как в соответствии с [6] минимальным функциональным элементом МАС является задача, то будем строить резервированную МАС на основе избыточности задач. Введем множество типов задач $TT = \{tt_i\}$ так, чтобы каждой задаче исходной МАС однозначно соответствовал один тип задачи. Введем множество задач резервированной МАС $RT = \{t_i\}$. Будем считать, что все задачи исходной МАС резервируются с одинаковой кратностью, тогда кратность резервирования задач определим отношением мощностей множеств задач резервированной МАС *RT* и типов задач *TT*.

Введем предикат, устанавливающий однозначное соответствие задачи резервированной МАС и ее типа:

$$confTypeTask : TT \times RT \rightarrow \{true, false\}, \quad (5)$$

где *TT* – множество типов задач, *RT* – множество задач резервированной МАС.

Аналогично введем избыточность исполнительных механизмов. Введем множество типов исполнительных механизмов $THWR = \{thwr_i\}$ так, чтобы каждому исполнительному механизму исходной МАС однозначно соответствовал один тип. Введем множество исполнительных механизмов резервированной МАС $RHWR = \{hwr_i\}$ и предикат, устанавливающий однозначное соответствие исполнительного механизма и его типа:

$$confTypeHwr : THWR \times RHWR \rightarrow \{true, false\}, \quad (6)$$

где *THWR* – множество типов исполнительных механизмов, *RHWR* – множество исполнительных механизмов резервированной МАС.

Так как стоимость резервирования исполнительного механизма может быть существенна, то кратность резервирования следует определять независимо для каждого типа исполнительного механизма. При введении избыточности задач можно полагать, что все задачи одного типа

являются эквивалентными, то есть при отказе определенной задачи может быть выполнена любая другая задача того же типа. Использование избыточности исполнительных механизмов в общем случае не допускает замену отказавшего исполнительного механизма любым другим того же типа, так как исполнительный механизм может быть частью конвейера, то есть результат его работы может быть использован для работы другого исполнительного механизма. Назовем исполнительным конвейером множество исполнительных механизмов системы, последовательно использующих результаты работы друг друга. Введем множество исполнительных конвейеров исходной МАС $PL = \{pl = \{hwr_i\}\}$. Так как отказ одного элемента конвейера фактически приводит к отказам всех исполнительных механизмов, образующих конвейер, то при введении избыточности следует резервировать одиночные исполнительные механизмы, не входящие в конвейеры, а также исполнительные конвейеры исходной МАС. Аналогично типизации исполнительных механизмов введем множество типов исполнительных конвейеров $TPL = \{tpl = \{tt_i\}\}$ в соответствии с множеством конвейеров исходной МАС. Введем множество исполнительных конвейеров резервированной МАС $RPL = \{pl = \{hwr_i\}\}$. Как было отмечено ранее, в силу существенной стоимости резервирования исполнительных механизмов, кратность резервирования следует определять независимо для каждого типа исполнительного конвейера и для каждого типа одиночного исполнительного механизма. Введем функцию, определяющую исполнительный конвейер, которому принадлежит исполнительный механизм:

$$PLofHWR(hwr) = \begin{cases} pl \Leftrightarrow \exists pl \in RPL : hwr \in pl \\ 0 \Leftrightarrow \forall pl \in RPL : hwr \in pl \end{cases}, \quad (7)$$

где hwr – исполнительный механизм, pl – исполнительный конвейер, RPL – множество исполнительный конвейеров резервированной МАС.

В исходной МАС необходимость использования исполнительных механизмов задана предикатом (3), который устанавливает соответствие необходимого исполнительного механизма конкретной задаче МАС. Так как в резервированной МАС введена избыточность как задач, так и исполнительных механизмов, то введем предикат, определяющий соответствие необходимого типа исполнительного механизма конкретному типу задачи:

$$reqThwrOfTt : THWR \times TT \rightarrow \{true, false\}, \quad (8)$$

где $THWR$ – множество типов исполнительных механизмов системы, TT – множество типов задач резервированной МАС.

Так как отказ агента вызывает отказы всех размещенных в нем задач, а отказ агентной платформы вызывает отказы всех задач всех агентов, размещенных в данной агентной платформе, то для обеспечения отказоустойчивости МАС необходимо также использовать избыточность агентов и агентных платформ. Использование резервирования таких компонентов МАС как задачи и исполнительные механизмы фактически добавляет в МАС резервные элементы, эквивалентные

уже существующим в исходной МАС. Так как агентная платформа предназначена для исполнения любых программных агентов, то будем считать, что агент – это программный компонент МАС, предназначенный для исполнения любых задач МАС. Тогда вместо использования резервирования, то есть введения агентов и агентных платформ, эквивалентных соответствующим компонентам исходной МАС, введем избыточное множество агентов резервированной МАС $RA = \{a_i\}$, в которых могут быть размещены задачи резервированной МАС, и избыточное множество агентных платформ $RHWP = \{hwp_i\}$, в которых могут быть размещены эти агенты. Условием размещения задач в агентах и агентов в агентных платформах является доступность необходимых для выполнения задач исполнительных механизмов. Для каждой агентной платформы резервированной МАС необходимо определить множество доступных ей исполнительных механизмов. В соответствии с разработанной формальной моделью МАС доступность исполнительного механизма агентной платформе определяется наличием связи между этими компонентами, однако если МАС обладает распределенной структурой, то исполнительные механизмы могут быть размещены в физически изолированных зонах, связь между которыми невозможна. Введем понятия локации как области физического размещения аппаратных компонентов МАС. Будем считать, что связь между агентной платформой и исполнительным механизмом возможна, только если они размещены в одной локации. Введем множество $L = \{l_i\}$ локаций резервированной МАС и предикат, определяющий принадлежность исполнительного механизма определенной локации:

$$confHwrL : RHWR \times L \to \{true, false\}, \tag{9}$$

где *RHWR* – множество исполнительных механизмов резервированной МАС, *L* – множество локаций.

Аналогично введем предикат, определяющий размещение агентных платформ по локациям:

$$confHwpL : RHWP \times L \to \{true, false\}, \tag{10}$$

где *RHWP* – множество агентных платформ резервированной МАС, *L* – множество локаций.

Для каждой агентной платформы можно определить множество потенциально доступных ей исполнительных механизмов, с которыми физически возможно реализовать связь

$$\begin{aligned} pHwr(hwp) = \{hwr \mid \exists l \in L : (confHwrL(hwr, l) = true) \wedge \\ (confHwpL(hwp, l) = true)\}, \end{aligned} \tag{11}$$

где *hwp* – агентная платформа, *hwr* – исполнительный механизм, *l* – локация, *L* – множество локаций резервированной МАС.

Будем считать, что в резервированной МАС реализована физическая связь между агентной платформой и некоторым подмножеством множества потенциально доступных ей исполнительных

механизмов. Переопределим предикат (4), чтобы задать доступность исполнительных механизмов агентным платформам в рамках резервированной МАС:

$$confHwrHwp : RHWR \times RHWP \rightarrow \{true, false\}, \quad (12)$$

где *RHWR* – множество исполнительных механизмов резервированной МАС, *RHWP* – множество агентных платформ.

Аналогично переопределим предикат (1), задающий распределение задач по агентам:

$$confTaskAgent : RT \times RA \rightarrow \{true, false\}, \quad (13)$$

где *RT* – множество задач резервированной МАС, *RA* – множество агентов.

Также переопределим предикат (2), задающий распределение агентов по агентным платформам:

$$confAgentHwp : RA \times RHWP \rightarrow \{true, false\}, \quad (14)$$

где *RA* – множество агентов резервированной МАС, *RHWP* – множество агентных платформ.

Разработанная формальная модель задает резервированную МАС как кортеж *RMAS* = <*RT, RA, RHWP, RHWR, RPL, L, TT, THWR, TPL*>, где *RT* – множество задач резервированной МАС, *RA* – множество агентов, *RHWP* – множество агентных платформ, *RHWR* – множество исполнительных механизмов, *RPL* – множество исполнительных конвейеров, *L* – множество локаций, *TT* – множество типов задач, *THWR* – множество типов исполнительных механизмов, *TPL* – множество типов исполнительных конвейеров. Соответствие задач и их типов задано предикатом (5), соответствие исполнительных механизмов и их типов – предикатом (6). Необходимость использования исполнительного механизма определенного типа задачей определенного типа задана предикатом (8). Физическое размещение агентных платформ и исполнительных механизмов в локациях задано соответственно предикатами (9) и (10). Доступность исполнительных механизмов для агентных платформ задана предикатом (12). Распределения задач по агентам и агентов по агентным платформам заданы предикатами (13) и (14) соответственно.

4. Условия работоспособности резервированной многоагентной системы

Логико-вероятностные методы основаны на использовании функций алгебры логики для формулировки условий работоспособности системы и переходе от логической функции работоспособности системы к вероятностным функциям, выражающим безотказность этой системы [9]. Так как задача является минимальным функциональным элементов МАС, то отказ одной из задач МАС приводит к отказу всей МАС в том случае, если данная задача должна быть выполнена. В случае распределенной программно-аппаратной МАС, в которой агенты выступают в качестве исполнительных контейнеров для задач, а агентные платформы являются в свою очередь контейнерами для агентов, отказ агента приводит к отказам всех принадлежащих ему задач, а отказ агентной платформе приводит к отказам задач, принадлежащих всем агентам,

размещенным в данной агентной платформе. Отказ исполнительного механизма МАС приводит к отказу всей МАС в том случае, если данный исполнительный механизм должен быть использован одной из задач, то есть, если одной из задач МАС необходим данный исполнительный механизм и данная задача должна быть выполнена. Кроме того, в случае существования исполнительных конвейеров отказ одного из элементов конвейера является причиной отказа всех остальных исполнительных механизмов данного конвейера. Таким образом, согласно разработанной модели резервированной распределенной программно-аппаратной МАС определим следующие отказы такой МАС: программный отказ задачи, программный отказ агента, аппаратный отказ агентной платформы, аппаратный отказ исполнительного механизма.

Логико-вероятностные методы расчета надежности технических систем выражают условия работоспособности системы в виде функции работоспособности системы, связывающей состояния элементов с состоянием системы [9]. Функция работоспособности системы может быть записана в виде дизъюнкции всем возможных кратчайших путей успешного функционирования, каждый из которых представляет собой такую конъюнкцию элементов системы, ни одну из компонент которой нельзя изъять, не нарушив функционирования системы [9]. Если МАС содержит интеллектуальных агентов, способных модифицировать условия выполнения своих действий, в частности в силу способности к обучению и адаптации, то на стадии проектирования невозможно задать условия запуска задач МАС. Так как согласно предложенному определению отказ отдельной задачи МАС приводит к отказу всей МАС только в том случае, если данная задача должна быть выполнена, то на стадии проектирования невозможно определить такой минимальный набор задач МАС, отказ одной из которых вызывает отказ всей МАС. Поэтому будем считать, что для успешного функционирования резервированной МАС необходимо существование по крайней мере одной работоспособной задачи каждого типа. Введем понятие минимальной функциональной конфигурации МАС как тройки:

$$MFC = < MT, MA, MHWP >, \qquad (15)$$

где *MT* – множество задач минимальной функциональной конфигурации, *MA* – множество агентов, *MHWP* – множество агентных платформ.

Множество задач минимальной функциональной конфигурации должно включать в себе одну и только одну задачу каждого типа из множества типов задач резервированной МАС *TT*, то есть для множества задач минимальной функциональной конфигурации должно быть выполнено условие:

$$(\forall tt \in TT : \exists! t \in MT : confTypeTask(tt, t) = true) \wedge (|TT| = |MT|), \qquad (16)$$

где *MT* – множество задач минимальной функциональной конфигурации, *TT* – множество типов задач резервированной МАС.

Так как для каждой задачи исходной МАС существует одна и только одна эквивалентная задача в множестве задач минимальной функциональной конфигурации, то будем считать, что для успешного функционирования резервированной МАС необходимо существование хотя бы одной минимальной функциональной конфигурации, все задачи которой находятся в работоспособном состоянии. Отметим, что отказ одной из задач минимальной функциональной конфигурации приводит к отказу всей МАС только в том случае, если выполнены условия запуска данной задачи, следовательно оценка надежности резервированной МАС, выполненная на основе предложенного определения, будет являться оценкой снизу. Так как для работоспособности отдельной задачи МАС необходима работоспособность агента, которому принадлежит данная задача, то для множества агентов минимальной функциональной конфигурации должно быть выполнено условие:

$$\forall t \in MT : \exists! a \in MA : confTaskAgent(t, a) = true, \quad (17)$$

где t – задача резервированной МАС, MT – множество задач минимальной функциональной конфигурации, a – агент резервированной МАС, MA – множество агентов минимальной функциональной конфигурации.

Так как для работоспособности агента МАС в свою очередь необходима работоспособность соответствующей агентной платформы, в которой данный агент размещен, то множество агентных платформ минимальной функциональной конфигурации должны быть сформировано исходя из следующего условия:

$$\forall a \in MA : \exists! p \in MHWP : confAgentHwp(a, p) = true, \quad (18)$$

где a – агент резервированной МАС, MA – множество агентов минимальной функциональной конфигурации, p – агентная платформа резервированной МАС, $MHWP$ – множество агентных платформ минимальной функциональной конфигурации.

Так как для функционирования отдельной задачи МАС может быть необходимо использование исполнительного механизма, то условия успешного функционирования резервированной МАС также должны включать условие работоспособности всех исполнительных механизмов, необходимых для исполнения задач МАС. Введем понятие минимальной работоспособной конфигурации как двойки:

$$MWC =< MFC, MWR >, \quad (19)$$

где MFC – минимальная функциональная конфигурации, MWR – множество исполнительных механизмов минимальной работоспособной конфигурации.

Так как минимальная функциональная конфигурации включает в себя задачи каждого типа, то в силу определения множеств типов задач и типов исполнительных механизмов резервированной МАС для исполнения всех задач минимальной функциональной конфигурации

необходимо использование ресурсов всех типов, то есть для множества исполнительных механизмов минимальной работоспособной конфигурации должно быть выполнено условие:

$$(\forall tr \in THWR : \exists! r \in MWR : confTypeHwr(tr, r) = true) \wedge (|THWR| = |MWR|), \quad (20)$$

где *tr* – тип исполнительного механизма резервированной МАС, *THWR* – множество типов исполнительных механизмов резервированной МАС, *r* – исполнительный механизм резервированной МАС, *MWR* – множество исполнительных механизмов минимальной работоспособной конфигурации.

Будем считать, что для успешного функционирования резервированной МАС необходимо и достаточно существование хотя бы одной минимальной работоспособной конфигурации, все элементы которой находятся в работоспособном состоянии. Так как доступность исполнительного механизма определяется наличием связи между агентной платформой и исполнительным механизмом в соответствии с предикатом (12), то для минимальной работоспособной конфигурации должно быть выполнено следующее условие:

$$\forall r \in MWR : \exists p \in MHWP : confHwrHwp(r, p) = true, \quad (21)$$

где *r* – исполнительный ресурс резервированной МАС, *MWR* – множество исполнительных ресурсов минимальной работоспособной конфигурации, *p* – агентная платформа резервированной МАС, *MHWP* – множество агентных платформ минимальной работоспособной конфигурации.

Так как исполнительные механизмы МАС, объединенные в конвейеры, последовательно используют результаты работы друг друга, то для успешного функционирования МАС необходимо, чтобы минимальная работоспособная конфигурация включала в себя исполнительные конвейеры каждого типа. Так как множество исполнительных механизмов минимальной работоспособной конфигурации включает в себя исполнительные механизмы всех типов, то для существования исполнительных конвейеров всех типов достаточно, чтобы для каждого исполнительного механизма минимальной работоспособной конфигурации, являющегося элементом конвейера, все остальные элементы данного конвейера также принадлежали данной минимальной работоспособной конфигурации. То есть для существования в минимальной работоспособной конфигурации исполнительных конвейеров всех типов достаточно выполнения следующего условия, сформулированного согласно формуле (7):

$$\forall r \in MWR : ((pl = PLofHWR(r) \neq 0) \Rightarrow (\forall pr \in pl : pr \in MWR)), \quad (22)$$

где *r, pr* – исполнительные механизмы резервированной МАС, *MWR* – множество исполнительных ресурсов минимальной работоспособной конфигурации, *pl* – исполнительный конвейер резервированной МАС.

Так как задача МАС может использовать только тот исполнительный механизм, который ей доступен в соответствии с предикатами (12), (13), (14), то минимальная работоспособная конфигурация должна удовлетворять следующему условию:

$$\forall t \in MT, \exists tt \in TT : confTypeTask(tt,t) = true, \forall tr : reqThwrOfTt(tr,tt) = true :$$
$$\exists a \in MA : confTaskAgent(t,a) = true, \exists p \in MHWP : confAgentHwp(a,p) = true, \quad (23)$$
$$\exists r \in MWR : (confTypeHwr(tr,r) = true) \wedge (confHwrHwp(r,p) = true),$$

где t – задача резервированной МАС, MT – множество задач минимальной работоспособной конфигурации, tt – тип задачи резервированной МАС, TT – множество типов задач резервированной МАС, tr – тип исполнительного механизма резервированной МАС, a – агент резервированной МАС, MA – множество агентов минимальной работоспособной конфигурации, p – агентная платформа резервированной МАС, $MHWP$ – множество агентных платформ минимальной работоспособной конфигурации, r – исполнительный механизм резервированной МАС, MWR – множество исполнительных механизмов минимальной работоспособной конфигурации.

Будем считать, что резервированная распределенная программно-аппаратная МАС функционирует успешно, если в МАС существует хотя бы одна минимальная работоспособная конфигурация (19), все элементы которой находятся в работоспособном состоянии и для которой выполнены условия (16) – (18) и (20) – (23). В соответствии с [9], определим функцию работоспособности резервированной МАС как дизъюнкцию всех кратчайших путей успешного функционирования, выраженных минимальными работоспособными конфигурациями:

$$\bigvee_{\forall MWC} \left[\left(\bigwedge_{\forall t \in MT} w(t) \right) \wedge \left(\bigwedge_{\forall a \in MA} w(a) \right) \wedge \left(\bigwedge_{\forall p \in MHWP} w(p) \right) \wedge \left(\bigwedge_{\forall r \in MWR} w(r) \right) \right], \quad (24)$$

где MWC – минимальная работоспособная конфигурация (19) резервированной МАС, t – задача резервированной МАС, MT – множество задач минимальной работоспособной конфигурации, $w()$ – функция алгебры логики, определяющая состояние компонента МАС, a – агент резервированной МАС, MA – множество агентов минимальной работоспособной конфигурации, p – агентная платформа резервированной МАС, $MHWP$ – множество агентных платформ минимальной работоспособной конфигурации, r – исполнительный механизм резервированной МАС, MWR – множество исполнительных механизмов минимальной работоспособной конфигурации.

5. Оценка надежности резервированной многоагентной системы

Согласно [10] функции алгебры логики, записанные в виде ортогональной дизъюнктивной нормальной формы, являются формами перехода к прямому замещению, то есть допускают одношаговое замещение логических переменных и логических операций на соответствующие вероятности и арифметические операции. Так как в соответствии с построением функция работоспособности резервированной МАС (24) является дизъюнктивной нормальной формой, то для перехода к вероятностной функции необходимо провести ее ортогонализацию.

Для получения функции работоспособности МАС, которая требуется для оценки надежности МАС с использованием логико-вероятностных методов, необходимо сформировать

множество всех минимальных работоспособных конфигураций резервированной МАС. Так как минимальные работоспособные конфигурации основаны на минимальных функциональных конфигурациях, то на первом этапе сформируем все множества задач *MT*, на базе которых могут быть построены минимальные функциональные конфигурации, то есть такие множества задач *MT*, для которых выполняется условие (16). Если считать, что элементы множества задач *MT* упорядочены, то такое множество представляет собой кортеж длины $N = |TT|$, где *TT* – множество типов задач резервированной МАС. В соответствии с (5) введем функцию, определяющую множество задач резервированной МАС конкретного типа:

$$tasksOfType(tt) = \{t \in RT \mid confTypeTask(tt, t) = true\}, \quad (25)$$

где *tt* – тип задачи резервированной МАС, *t* – задача резервированной МАС, *RT* – множество задач резервированной МАС.

Каждый элемент кортежа *MT* должен соответствовать определенному типу задачи резервированной МАС и принимать значения из множества задач резервированной МАС данного типа, определенного в соответствии с формулой (25). Тогда множество множеств задач *MT* получим как множество всех возможных кортежей, то есть как декартово произведение множеств задач каждого типа:

$$\tilde{MT} = \{MT = (t_1, ..., t_n) \mid n = |TT|, \forall i : \exists! tt_i \in TT : t_i \in tasksOfType(tt_i)\}, \quad (26)$$

где t_i – задача резервированной МАС, *TT* – множество типов задач резервированной МАС, tt_i – тип задачи резервированной МАС.

Для каждого из полученных множеств задач *MT* сформируем минимальную функциональную конфигурацию, определив множество агентов, необходимых для исполнения задач из *MT*, в соответствии с (17) и множество агентных платформ, необходимых для исполнения соответствующих агентов, в соответствии с (18), тогда множество всех минимальных функциональных конфигураций определено следующим образом:

$$\tilde{MFC} = \{(MT, MA, MHWP) \mid MT \in \tilde{MT}\}, \quad (27)$$

где *(MT, MA, MHWP)* – минимальная функциональная конфигурация в соответствии с определением (15) и условиями (16) – (18), $\tilde{MT}$ – множество множеств задач, для которых выполнено условие (16).

Для каждой из полученных минимальных функциональных конфигураций могут быть сформированы соответствующие минимальные работоспособные конфигурации. Минимальная работоспособная конфигурация должна удовлетворять условию (23) доступности необходимых для выполнения задач ресурсов. Так как потенциальная доступность исполнительных механизмов агентным платформам определяется принадлежностью исполнительного механизма и агентной платформы одной и той же локации в соответствии с функцией (11), то исключим те минимальные

функциональные конфигурации, при использовании которых для построения минимальных работоспособных конфигураций заведомо не будет выполнено условие (23). Такие минимальные функциональные конфигурации определим исходя из условия наличия двух задач, расположенных в разных локациях и требующих для своего выполнения один и тот же тип исполнительного механизма. Так как минимальная работоспособная конфигурация в соответствии с (20) включает в себя только один исполнительный механизм каждого типа, то для одной из задач, расположенных в разных локациях, условие (24) доступности необходимых исполнительных механизмов будет нарушено. В соответствии с предикатами (10), (13), (14) введем функцию, определяющую локацию, в которой расположена задача, следующим образом:

$$lOfT(t) = l \in L \Leftrightarrow (\exists a \in RA : confTaskAgent(t,a) = true,$$
$$\exists p \in RHWP : (confAgentHwp(a,p) = true) \wedge (confHwpL(p,l) = true)),$$

где t – задача резервированной МАС, l – локация, L – множество локаций, a – агент резервированной МАС, RA – множество агентов резервированной МАС, p – агентная платформа резервированной МАС, $RHWP$ – множество агентных платформ резервированной МАС.

Условие исключения минимальной функциональной конфигурации сформулируем следующим образом:

$$\exists t_1 \in MT, \exists t_2 \in MT, (\exists tt_1 \in TT : confTypeTask(tt_1, t_1) = true),$$
$$(\exists tt_2 \in TT : confTypeTask(tt_2, t_2) = true), \exists rt \in THWR : \quad (28)$$
$$(lOfT(t_1) \neq lOfT(t_2)) \wedge (rt \in reqThwrOfTt(tt_1) \wedge (rt \in reqThwrOfTt(tt_2)),$$

где t_1, t_2 – задачи резервированной МАС, MT – множество задач минимальной функциональной конфигурации, tt_1, tt_2 – типы задач резервированной МАС, TT – множество типов задач резервированной МАС, rt – тип исполнительного механизма резервированной МАС, $THWR$ – множество типов исполнительных механизмов резервированной МАС.

Исключая из множества всех минимальных функциональных конфигураций (27) такие конфигурации, для которых выполнено условие (28), сформируем множество $\tilde{MFC}'$.

Для каждой минимальной функциональной конфигурации из множества (27), для которой не выполнено условие (28), сформируем множество всех возможных минимальных работоспособных конфигураций. Для этого определим все множества исполнительных механизмов MWR, которые удовлетворяют условиям (20) и (21). Так как в соответствии с условием (20) множество исполнительных механизмов MWR содержит только один исполнительный механизм каждого типа, то если считать, что элементы множества исполнительных механизмов MWR упорядочены, то такое множество представляет собой кортеж длины $N = |THWR|$, где $THWR$ – множество типов исполнительных механизмов резервированной МАС. В соответствии с (6) введем функцию, определяющую множество исполнительных механизмов резервированной МАС конкретного типа:

$$hwrOfType(rt) = \{hwr \in RHWR \mid confTypeHwr(rt, hwr) = true\}, \quad (29)$$

где *rt* – тип исполнительного механизма резервированной МАС, *hwr* – исполнительный механизм резервированной МАС, *RHWR* – множество исполнительных механизмов резервированной МАС.

В соответствии с (12) и (29) введем функцию, определяющую множество исполнительных механизмов определенного типа, доступных агентным платформам минимальной функциональной конфигурации:

$$avHwrOfType(rt, MFC) = \{hwr \in RHWR \mid hwr \in hwrOfType(rt) \wedge (\exists p \in MHWP \in MFC : confHwrHwp(hwr, p) = true)\}, \quad (30)$$

где *rt* – тип исполнительного механизма резервированной МАС, *MFC* – минимальная функциональная конфигурация, *hwr* – исполнительный механизм резервированной МАС, *RHWR* – множество исполнительных механизмов резервированной МАС, *p* – агентная платформа резервированной МАС, *MHWP* – множество агентных платформ минимальной функциональной конфигурации.

Каждый элемент кортежа *MWR* должен соответствовать определенному типу исполнительного механизма резервированной МАС и в соответствии с условием (21) принимать значения из определенного в соответствии с формулой (30) множества исполнительных механизмов резервированной МАС данного типа, доступных агентным платформам минимальной функциональной конфигурации. Тогда искомое множество множеств исполнительных механизмов *MWR* получим как множество всех возможных кортежей, то есть как декартово произведение множеств исполнительных механизмов резервированной МАС каждого типа, доступных агентным платформам минимальной функциональной конфигурации:

$$\tilde{MWR} = \{MWR = (r_1, ..., r_m) \mid m = |THWR|, \forall i : \exists! rt_i \in THWR : r_i \in avHwrOfType(rt_i, MFC)\}, \quad (31)$$

где r_i – исполнительный механизм резервированной МАС, *THWR* – множество типов исполнительных механизмов резервированной МАС, rt_i – тип исполнительного механизма резервированной МАС, *MFC* – минимальная функциональная конфигурация.

Исключая из полученного множества (31) такие множества исполнительных механизмов, для которых не выполнены условие (22) существования исполнительных конвейеров каждого типа и условие (23) доступности всех исполнительных механизмов, необходимых задачам минимальной функциональной конфигурации, сформируем множество $\tilde{MWR}'$. Множество минимальных работоспособных конфигураций, основанных на минимальной функциональной конфигурации *MFC*, определим на основе полученного множества $\tilde{MWR}'$:

$$\tilde{MWC} = \{(MFC, mwr) \mid mwr \in \tilde{MWR}'\}, \quad (32)$$

где *MFC* – минимальная функциональная конфигурация, *mwr* – множество исполнительных механизмов.

Разработанная методика оценки надежности резервированной распределенной программно-аппаратной МАС состоит из следующих этапов:

1) определение всех возможных множеств задач *MT*, на основе которых могут быть построены минимальные функциональные конфигурации, в соответствии с (26);

2) формирование множества $\tilde{MFC}$ всех возможных минимальных функциональных конфигурация на основе множеств задач *MT*, полученных на этапе 1, в соответствии с (27);

3) формирование множества $\tilde{MFC}'$ всех допустимых минимальных функциональных конфигураций путем исключения из множества $\tilde{MFC}$, полученного на этапе 2, минимальных функциональных конфигураций, удовлетворяющих условию (28);

4) для каждой минимальной функциональной конфигурации из множества $\tilde{MFC}'$, полученного на этапе 3:

4.1) формирование в соответствии с (31) множества $\tilde{MWR}$ всех возможных множеств доступных минимальной функциональной конфигурации исполнительных механизмов, удовлетворяющих условию (20) включения единственного исполнительного механизма каждого типа, на основе которых могут быть построены минимальные работоспособные конфигурации;

4.2) формирование множества $\tilde{MWR}'$ путем исключения из множества $\tilde{MWR}$, полученного на шаге 4.1, таких множеств исполнительных механизмов, для которых не выполнены (22) существования исполнительных конвейеров каждого типа и условие (23) доступности всех исполнительных механизмов, необходимых задачам минимальной функциональной конфигурации;

4.3) формирование множества $\tilde{MWC}$ допустимых минимальных работоспособных конфигураций в соответствии с (32) на основе множества $\tilde{MWR}'$, полученного на этапе 4.2;

5) формирование функции работоспособности резервированной МАС в соответствии с (24) на основе всех допустимых минимальных работоспособных конфигураций, сформированных на этапе 4;

6) ортогонализация функции работоспособности резервированной МАС, полученной на этапе 5, в соответствии с разложением Шеннона [9];

7) формирование вероятностной функции работоспособности резервированной МАС в соответствии с логико-вероятностным методом [9] путем замещения логических переменных

соответствующими вероятностями их безотказной работы, а логических операций соответствующими арифмитическими операциями.

6. Вычислительный эксперимент

Для подтверждения работоспособности предложенной авторами методики оценки надежности резервированных распределенных программно-аппаратных МАС был проведен вычислительный эксперимент. Аналитическое выражение для вероятности безотказной работы резервированной МАС на интервале времени было получено с использованием разработанной методики. Для получения экспериментальных значений вероятности безотказной работы МАС был применен метод статистического моделирования.

Рассмотрим методику получения экспериментальных значений вероятности безотказной работы МАС на интервале времени *[0; t]*, состоящую из N испытаний. Целью каждого испытания является определение состояния МАС на основе состояний всех ее элементов. Если в течение N испытаний работоспособное состояние МАС было зафиксировано N_+ раз, то согласно статистическому определению вероятности, вероятность безотказной работы МАС на интервале времени *[0; t]* можно определить как отношение числа успешных испытаний N_+ к общему числу испытаний N. Каждое испытание состоит из нескольких этапов. Первый этап испытания заключается в моделировании случайной величины наработки на отказ для каждого из компонентов резервированной МАС согласно заданному закону распределения. На втором этапе испытания определяем состояние каждого из каждого системы исходя из полученных наработок на отказ. Компонент системы считается работоспособным, если соответствующая наработка на отказ превышает величину заданного интервала времени. На третьем этапе испытания проводим модификацию определенных ранее состояний компонентов системы на основе следующих правил:

- если зафиксирован отказ агентной платформы, то фиксируем отказ всех размещенных в ней агентов, при этом множество таких агентов определяем согласно предикату (14);
- если зафиксирован отказ агента, то фиксируем отказ всех его задач, при этом множество таких задач определяем согласно предикату (13);
- если зафиксирован отказ исполнительного механизма r, входящего в конвейер, то фиксируем отказ всех исполнительных механизмов, образующих данный конвейер $\forall pr \in PLofHWR(r)$;
- если в соответствии с предикатом (12) все агентные платформы, которым доступен исполнительный механизм, находятся в состоянии отказа, то фиксируем отказ данного исполнительного механизма.

На четвертом этапе испытания, исключая из соответствующих множеств компонентов системы те компоненты, которые находятся в состоянии отказа, формируем следующие множества работоспособных компонентов системы: RT' – множество работоспособных задач, RA' – множество

работоспособных агентов, *RHWP'* – множество работоспособных агентных платформ, *RHWR'* – множество работоспособных исполнительных механизмов, *RPL'* – множество работоспособных исполнительных конвейеров. На пятом этапе испытания проводим проверку выполнения необходимых условий работоспособности МАС:

- множество работоспособных задач содержит хотя бы одну задачу каждого типа;
- множество работоспособных исполнительных механизмов содержит хотя бы один исполнительный механизм каждого типа;
- множество работоспособных исполнительных конвейеров содержит хотя бы один исполнительный конвейер каждого типа.

Если хотя бы одно из условий не выполнено, то фиксируем отказ всей резервированной МАС и переходим к следующему испытанию. В противном случае на шестом этапе испытания проводим проверку достаточных условий работоспособности МАС в соответствии с условиями, сформулированными в разделе 4.

Для проведения вычислительного эксперимента был разработан комплекс программных средств на языке программирования C++, реализующий следующую функциональность:

- получение экспериментальных значений вероятности безотказной работы резервированной МАС в соответствии с методикой, приведенной в данном разделе;
- получение аналитического выражения для вероятности безотказной работы резервированной МАС на интервале времени *[0; t]* в соответствии с методикой, описанной в разделе 5.

Описанный эксперимент был проведен для следующей тестовой МАС, заданной разработанной авторами моделью:

- множество типов задач $TT = \{tt_1, tt_2, tt_3\}$;
- множество задач $RT = \{t_{11}, t_{12}, t_{21}, t_{22}, t_{31}, t_{32}\}$;
- предикат (5), устанавливающий соответствие задачи и ее типа, принимает истинное значение на множестве $\{(tt_1, t_{11}), (tt_1, t_{12}), (tt_2, t_{21}), (tt_2, t_{22}), (tt_3, t_{31}), (tt_3, t_{32})\}$;
- множество типов исполнительных механизмов $THWR = \{rt_1, rt_2, rt_3\}$;
- множество исполнительных механизмов $RHWR = \{r_{11}, r_{12}, r_{21}, r_{22}, r_{31}, r_{32}\}$;
- предикат (6), устанавливающий соответствие исполнительного механизма и его типа, принимает истинное значение на множестве $\{(rt_1, r_{11}), (rt_1, r_{12}), (rt_2, r_{21}), (rt_2, r_{22}), (rt_3, r_{31}), (rt_3, r_{32})\}$;
- множество типов исполнительных конвейеров $TPL = \{plt_1\}$;
- множество исполнительных конвейеров $RPL = \{pl_1 = \{r_{21}, r_{31}\}, pl_2 = \{r_{22}, r_{32}\}\}$;
- множество агентов $RA = \{a_1, a_2, a_3, a_4\}$;
- предикат (13), задающий распределение задач по агентам, принимает истинное значение на множестве $\{(t_{11}, a_1), (t_{32}, a_1), (t_{12}, a_2), (t_{31}, a_2), (t_{21}, a_3), (t_{22}, a_4)\}$;
- множество агентных платформ $RHWP = \{h_1, h_2, h_3\}$;

- предикат (14), задающий распределение агентов по агентным платформам, принимает истинное значение на множестве $\{(a_1, h_1), (a_2, h_2), (a_3, h_2), (a_4, h_3)\}$;
- множество локаций $L = \{l_1, l_2\}$;
- предикат (9), определяющий размещение исполнительных механизмов по локациям, принимает истинное значение на множестве $\{(r_{11}, l_1), (r_{12}, l_2), (r_{21}, l_2), (r_{22}, l_2), (r_{31}, l_2), (r_{32}, l_2)\}$;
- предикат (10), определяющий размещение агентных платформ по локациям, принимает истинное значение на множестве $\{(h_1, l_1), (h_2, l_2), (h_3, l_3)\}$;
- предикат (12), определяющий доступность исполнительных механизмов для агентных платформ, принимает истинное значение на следующем множестве $\{(r_{11}, h_1), (r_{12}, h_2), (r_{21}, h_2), (r_{21}, h_3), (r_{22}, h_2), (r_{22}, h_3), (r_{31}, h_2), (r_{31}, h_3), (r_{32}, h_2), (r_{32}, h_3)\}$;
- предикат (8), определяющий соответствие необходимого типа исполнительного механизма конкретному типу задачи, принимает истинное значение на множестве $\{(rt_1, tt_1), (rt_1, tt_3), (rt_2, tt_2), (rt_3, tt_2)\}$.

Графически конфигурация тестовой резервированной МАС представлена на рисунке 1.

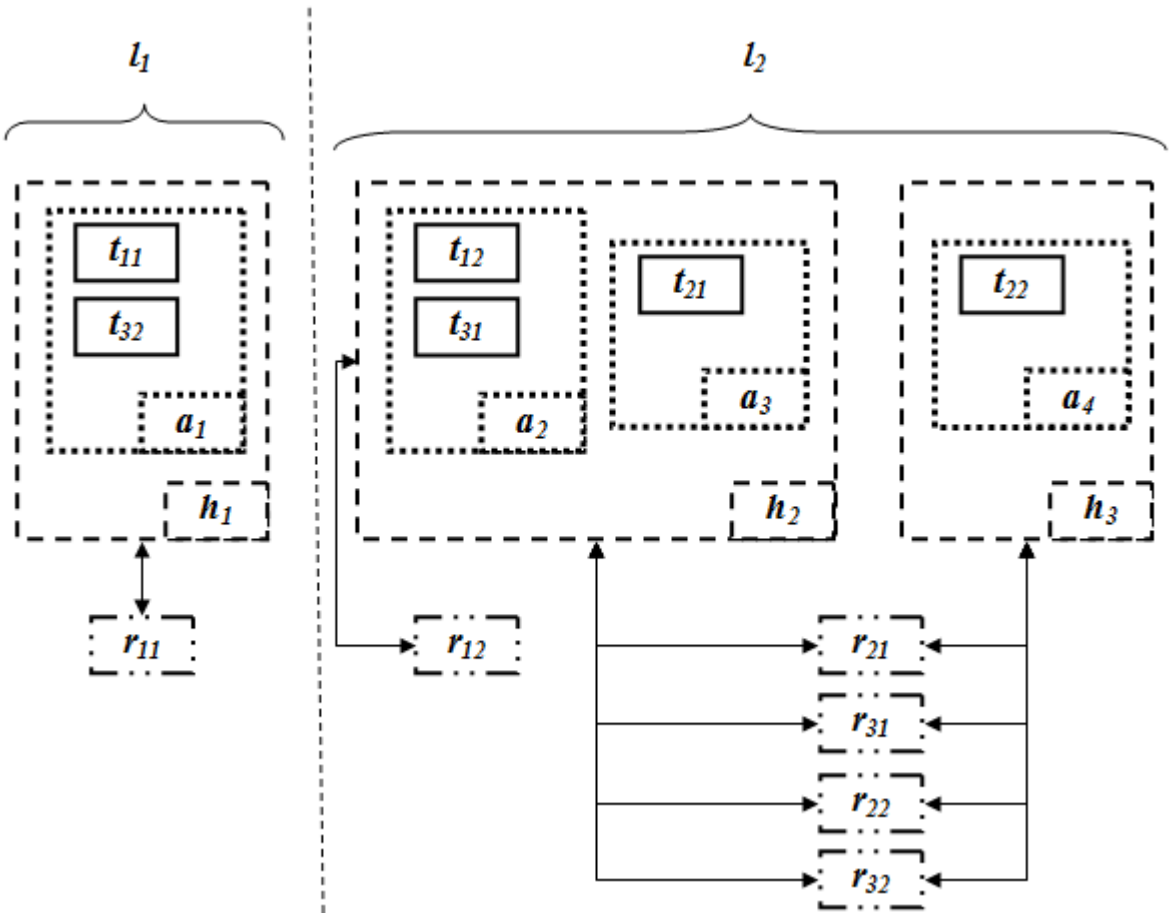


Рисунок 1 – Конфигурация тестовой резервированной МАС

Рассмотрим получение аналитической функции надежности (вероятности безотказной работы) для тестовой резервированной МАС в соответствии с разработанной методикой по этапам.

Сформированное на этапе 1 множество всех возможных множеств задач *MT*, на основе которых могут быть построены минимальные функциональные конфигурации, представлено в таблице 1.

Таблица 1 – Множество всех множеств задач MT (этап 1)

Множество *MT*	**Задача типа tt_1**	**Задача типа tt_2**	**Задача типа tt_3**
MT_1	t_{11}	t_{21}	t_{31}
MT_2	t_{11}	t_{21}	t_{32}
MT_3	t_{11}	t_{22}	t_{31}
MT_4	t_{11}	t_{22}	t_{32}
MT_5	t_{12}	t_{21}	t_{31}
MT_6	t_{12}	t_{21}	t_{32}
MT_7	t_{12}	t_{22}	t_{31}
MT_8	t_{12}	t_{22}	t_{32}

Сформированное на этапе 2 множество $\tilde{MFC}$ всех возможных минимальных функциональных конфигураций *MFC* на основе множеств задач *MT*, полученных на этапе 1, представлено в таблице 2.

Таблица 2 – Множество всех возможных минимальных функциональных конфигураций (этап 2)

Множество *MFC*	**Множество *MT***	**Множество *MA***	**Множество *MHWP***
MFC_1	$\{t_{11}, t_{21}, t_{31}\}$	$\{a_1, a_2, a_3\}$	$\{h_1, h_2\}$
MFC_2	$\{t_{11}, t_{21}, t_{32}\}$	$\{a_1, a_3\}$	$\{h_1, h_2\}$
MFC_3	$\{t_{11}, t_{22}, t_{31}\}$	$\{a_1, a_2, a_4\}$	$\{h_1, h_2, h_3\}$
MFC_4	$\{t_{11}, t_{22}, t_{32}\}$	$\{a_1, a_4\}$	$\{h_1, h_3\}$
MFC_5	$\{t_{12}, t_{21}, t_{31}\}$	$\{a_2, a_3\}$	$\{h_2\}$
MFC_6	$\{t_{12}, t_{21}, t_{32}\}$	$\{a_1, a_2, a_3\}$	$\{h_1, h_2\}$
MFC_7	$\{t_{12}, t_{22}, t_{31}\}$	$\{a_2, a_4\}$	$\{h_2, h_3\}$
MFC_8	$\{t_{12}, t_{22}, t_{32}\}$	$\{a_1, a_2, a_4\}$	$\{h_1, h_2, h_3\}$

На третьем этапе из сформированного множества $\tilde{MFC}$ исключаем такие минимальные функциональные конфигурации, для которых выполнено условие (28), то есть те конфигурации, в которых существуют расположенные в разных локациях задачи, которым необходим

исполнительный механизм одного типа. Для тестовой МАС из множества $\tilde{MFC}$ исключаем следующие минимальные функциональные конфигурации:

- MFC_1 и MFC_3, так как задача t_{11} типа tt_1 требует исполнительный механизм типа rt_1, а задача t_{31} типа tt_3 также требует исполнительный механизм типа rt_1, при этом задача t_{11} расположена в локации l_1, а задача t_{31} расположена в локации l_2;
- MFC_6 и MFC_8, так как задача t_{12} типа tt_1 требует исполнительный механизм типа rt_1, а задача t_{32} типа tt_3 также требует исполнительный механизм типа rt_1, при этом задача t_{12} расположена в локации l_2, а задача t_{32} расположена в локации l_1.

Рассмотрим четвертый этап методики для допустимой минимальной функциональной конфигурации MFC_2. Сформированное на этапе 4.1. множество $\tilde{MWR}$ всех возможных множеств доступных минимальной функциональной конфигурации MFC_2 исполнительных механизмов, удовлетворяющих условию (20) включения единственного исполнительного механизма каждого типа, представлено в таблице 3.

Таблица 3 – Множество всех множеств исполнительных механизмов MWR (этап 4.1 для MFC_2)

Множество *MWR*	**Исполнительный механизм типа rt_1**	**Исполнительный механизм типа rt_2**	**Исполнительный механизм типа rt_3**
MWR_1	r_{11}	r_{21}	r_{31}
MWR_2	r_{11}	r_{21}	r_{32}
MWR_3	r_{11}	r_{22}	r_{31}
MWR_4	r_{11}	r_{22}	r_{32}
MWR_5	r_{12}	r_{21}	r_{31}
MWR_6	r_{12}	r_{21}	r_{32}
MWR_7	r_{12}	r_{22}	r_{31}
MWR_8	r_{12}	r_{22}	r_{32}

На этапе 4.2 исключаем из сформированного множества $\tilde{MWR}$ такие множества исполнительных механизмов *MWR*, для которых не выполнены (22) существования исполнительных конвейеров каждого типа и условие (23) доступности всех исполнительных механизмов, необходимых задачам минимальной функциональной конфигурации. Для минимальной функциональной конфигурации MFC_2 исключаем следующие множества *MWR*:

- MWR_2 и MWR_6, так как исполнительный механизм r_{21} входит в исполнительный конвейер $pl_1 = \{r_{21}, r_{31}\}$, а исполнительный механизм r_{31} данного конвейера не входит в рассматриваемые множества *MWR*;

- MWR_3 и MWR_7, так как исполнительный механизм r_{22} входит в исполнительный конвейер $pl_2 = \{r_{22}, r_{32}\}$, а исполнительный механизм r_{32} данного конвейера не входит в рассматриваемые множества *MWR*.

Сформированное на этапе 4.3 множество всех допустимых минимальных работоспособных конфигураций $\tilde{MWC}$, основанных на MFC_2, представлено в таблице 4.

Таблица 4 – Множество всех допустимых минимальных работоспособных конфигураций (этап 4.3 для MFC_2)

Множество *MWC*	**Множество *MT***	**Множество *MA***	**Множество *MHWP***	**Множество *MWR***
MWC_{21}	$\{t_{11}, t_{21}, t_{32}\}$	$\{a_1, a_3\}$	$\{h_1, h_2\}$	$\{r_{11}, r_{21}, r_{31}\}$
MWC_{22}	$\{t_{11}, t_{21}, t_{32}\}$	$\{a_1, a_3\}$	$\{h_1, h_2\}$	$\{r_{11}, r_{22}, r_{32}\}$
MWC_{23}	$\{t_{11}, t_{21}, t_{32}\}$	$\{a_1, a_3\}$	$\{h_1, h_2\}$	$\{r_{12}, r_{21}, r_{31}\}$
MWC_{24}	$\{t_{11}, t_{21}, t_{32}\}$	$\{a_1, a_3\}$	$\{h_1, h_2\}$	$\{r_{12}, r_{22}, r_{32}\}$

Сформированные в соответствии с четвертым этапом методики множества всех допустимых минимальных работоспособных конфигураций, основанных на допустимых минимальных функциональных конфигурациях MFC_4, MFC_5 и MFC_7, представлены в таблицах 5 – 7 соответственно.

Таблица 5 – Множество всех допустимых минимальных работоспособных конфигураций (этап 4 для MFC_4)

Множество *MWC*	**Множество *MT***	**Множество *MA***	**Множество *MHWP***	**Множество *MWR***
MWC_{41}	$\{t_{11}, t_{22}, t_{32}\}$	$\{a_1, a_4\}$	$\{h_1, h_3\}$	$\{r_{11}, r_{21}, r_{31}\}$
MWC_{42}	$\{t_{11}, t_{22}, t_{32}\}$	$\{a_1, a_4\}$	$\{h_1, h_3\}$	$\{r_{11}, r_{22}, r_{32}\}$

Таблица 6 – Множество всех допустимых минимальных работоспособных конфигураций (этап 4 для MFC_5)

Множество *MWC*	**Множество *MT***	**Множество *MA***	**Множество *MHWP***	**Множество *MWR***
MWC_{51}	$\{t_{12}, t_{21}, t_{31}\}$	$\{a_2, a_3\}$	$\{h_2\}$	$\{r_{12}, r_{21}, r_{31}\}$
MWC_{52}	$\{t_{12}, t_{21}, t_{31}\}$	$\{a_2, a_3\}$	$\{h_2\}$	$\{r_{12}, r_{22}, r_{32}\}$

Таблица 7 – Множество всех допустимых минимальных работоспособных конфигураций (этап 4 для MFC_7)

Множество *MWC*	**Множество *MT***	**Множество *MA***	**Множество *MHWP***	**Множество *MWR***
MWC_{71}	$\{t_{12}, t_{22}, t_{31}\}$	$\{a_2, a_4\}$	$\{h_2, h_3\}$	$\{r_{12}, r_{21}, r_{31}\}$
MWC_{72}	$\{t_{12}, t_{22}, t_{31}\}$	$\{a_2, a_4\}$	$\{h_2, h_3\}$	$\{r_{12}, r_{22}, r_{32}\}$

На пятом этапе методики формируем функцию работоспособности всей резервированной МАС в соответствии с (24):

$$\begin{aligned}
&w(t_{11})w(t_{21})w(t_{32})w(a_1)w(a_3)w(h_1)w(h_2)w(r_{11})w(r_{21})w(r_{31}) + \\
&w(t_{11})w(t_{21})w(t_{32})w(a_1)w(a_3)w(h_1)w(h_2)w(r_{11})w(r_{22})w(r_{32}) + \\
&w(t_{11})w(t_{21})w(t_{32})w(a_1)w(a_3)w(h_1)w(h_2)w(r_{12})w(r_{21})w(r_{31}) + \\
&w(t_{11})w(t_{21})w(t_{32})w(a_1)w(a_3)w(h_1)w(h_2)w(r_{12})w(r_{22})w(r_{32}) + \\
&w(t_{11})w(t_{22})w(t_{32})w(a_1)w(a_4)w(h_1)w(h_3)w(r_{11})w(r_{21})w(r_{31}) + \\
&w(t_{11})w(t_{22})w(t_{32})w(a_1)w(a_4)w(h_1)w(h_3)w(r_{11})w(r_{22})w(r_{32}) + \\
&w(t_{12})w(t_{21})w(t_{31})w(a_2)w(a_3)w(h_2)w(r_{12})w(r_{21})w(r_{31}) + \\
&w(t_{12})w(t_{21})w(t_{31})w(a_2)w(a_3)w(h_2)w(r_{12})w(r_{22})w(r_{32}) + \\
&w(t_{12})w(t_{22})w(t_{31})w(a_2)w(a_4)w(h_2)w(h_3)w(r_{12})w(r_{21})w(r_{31}) + \\
&w(t_{12})w(t_{22})w(t_{31})w(a_2)w(a_4)w(h_2)w(h_3)w(r_{12})w(r_{22})w(r_{32}),
\end{aligned} \tag{33}$$

где t_i – задача тестовой резервированной МАС, a_i – агент тестовой резервированной МАС, h_i – агентная платформа тестовой резервированной МАС, r_i – исполнительный механизм тестовой резервированной МАС, $w()$ – функция алгебры логики, определяющая состояние компонента тестовой МАС.

На шестом этапе производим ортогонализация функции работоспособности тестовой МАС (33). Будем считать, что наработка на отказ всех компонентов тестовой МАС распределена экспоненциально, при этом отказы всех задач тестовой МАС происходят с интенсивностью $\lambda_t = 0.01$, отказы всех агентов – с интенсивностью $\lambda_a = 0.009$, отказы всех агентных платформ – с интенсивностью $\lambda_p = 0.007$, а отказы всех исполнительных механизмов – с интенсивностью $\lambda_r = 0.005$. На седьмом этапе методики, произведя замену логических переменных соответствующими вероятностями и арифметических операций логическими в соответствии с логико-вероятностным методом [9], получаем аналитическое выражение для вероятности безотказной работы тестовой МАС:

$$\begin{aligned}
&PA(t) = 8p_t^3(t)p_a^2(t)p_h^2(t)p_r^3(t) - 2p_t^3(t)p_a^2(t)p_h^2(t)p_r^4(t) - \\
&4p_t^3(t)p_a^2(t)p_h^2(t)p_r^5(t) + p_t^3(t)p_a^2(t)p_h^2(t)p_r^6(t) + \\
&p_t^4(t)p_a^3(t)p_h^2(t)p_r^5(t) + p_t^4(t)p_a^3(t)p_h^3(t)p_r^5(t) - \\
&2p_t^4(t)p_a^3(t)p_h^2(t)p_r^3(t) - 2p_t^4(t)p_a^3(t)p_h^3(t)p_r^3(t) -
\end{aligned} \tag{34}$$

$p_t^6(t)p_a^4(t)p_h^3(t)p_r^6(t) + p_t^5(t)p_a^3(t)p_h^3(t)p_r^6(t) +$
$p_t^5(t)p_a^3(t)p_h^2(t)p_r^5(t) + 2p_t^6(t)p_a^4(t)p_h^3(t)p_r^4(t) -$
$2p_t^5(t)p_a^3(t)p_h^3(t)p_r^4(t) - 2p_t^5(t)p_a^3(t)p_h^2(t)p_r^3(t) -$
$p_t^3(t)p_a^2(t)p_h(t)p_r^5(t) + 2p_t^3(t)p_a^2(t)p_h(t)p_r^3(t),$

где $p_t(t) = e^{-\lambda_t t}$ – функция надежности задачи тестовой МАС, $p_a(t) = e^{-\lambda_a t}$ – функция надежности агента тестовой МАС, $p_h(t) = e^{-\lambda_h t}$ – функция надежности агентной платформы тестовой МАС, $p_r(t) = e^{-\lambda_r t}$ – функция надежности исполнительного механизма тестовой МАС.

Для получения статистических значений вероятности безотказной работы тестовой МАС на интервале времени *[0; t]* были проведены 10 вычислительных экспериментов, состоящих из 1000 испытаний, в соответствии с методикой получения экспериментальных значений, описанной в настоящем разделе. Теоретические значения *PA(t)* вероятностей безотказной работы тестовой МАС на интервалах времени *[0; t]* для *t = 1…52*, рассчитанные в соответствии с аналитической функцией надежности (34), и полученные экспериментальные значения *PS(t)* приведены в таблице 8.

Таблица 8 – Теоретические и экспериментальные значения вероятности безотказной работы тестовой МАС

Время t	**Вероятность безотказной работы МАС PA(t)**	**Вероятность безотказной работы МАС PS(t)**	**Погрешность**	
			Абсолютная	**Относительная**
1	0,997770	0,998000	-0,000230	-0,000231
4	0,968081	0,968800	-0,000719	-0,000743
7	0,912857	0,914300	-0,001443	-0,001581
10	0,841828	0,842500	-0,000672	-0,000798
13	0,762579	0,764000	-0,001421	-0,001863
16	0,680749	0,684300	-0,003551	-0,005216
19	0,600332	0,606000	-0,005668	-0,009441
22	0,523987	0,523500	0,000487	0,000929
25	0,453344	0,465800	-0,012456	-0,027476
28	0,389262	0,387400	0,001862	0,004783
31	0,332044	0,327600	0,004444	0,013384
34	0,281610	0,282100	-0,000490	-0,001740

37	0,237629	0,233700	0,003929	0,016534
40	0,199621	0,196600	0,003021	0,015134
43	0,167027	0,164900	0,002127	0,012734
46	0,139260	0,136100	0,003160	0,022691
49	0,115742	0,113000	0,002742	0,023691
52	0,095922	0,095300	0,000622	0,006484

График функции надежности тестовой МАС (34) с отмеченными экспериментальными значениями вероятностей безотказной работы представлен на рисунке 2.

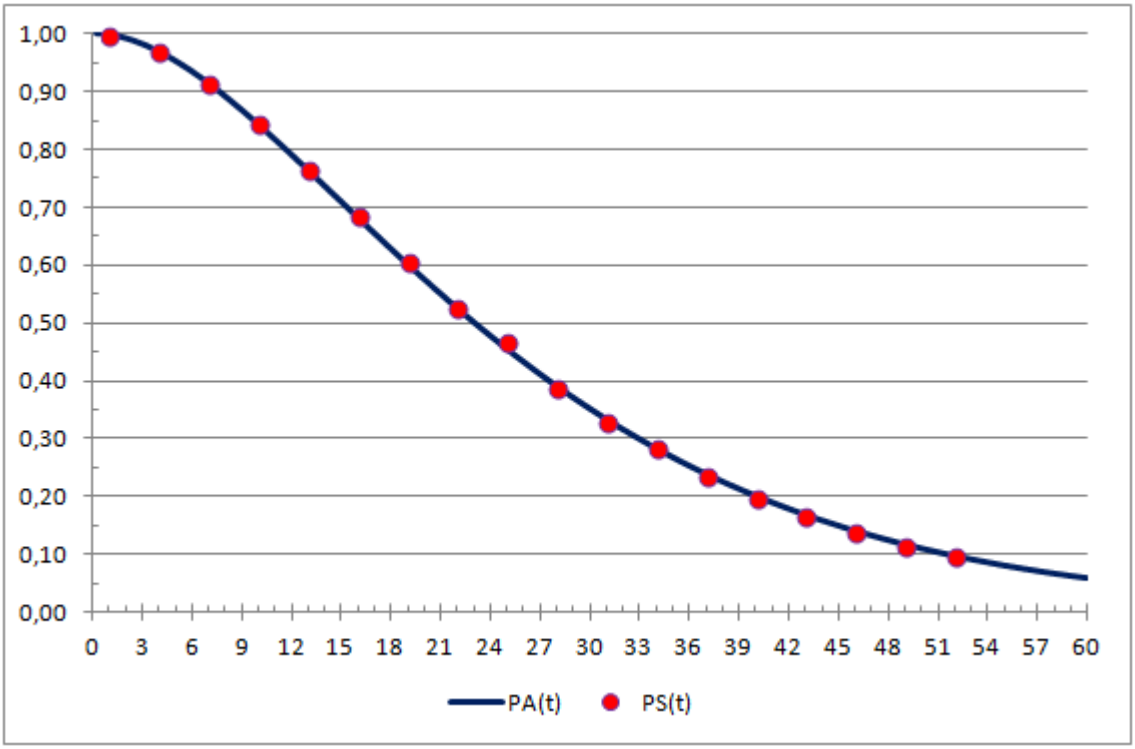


Рисунок 2 – Функция надежности тестовой МАС PA(t) и экспериментальные значения вероятности безотказной работы PS(t)

Проведенный эксперимент подтверждает работоспособность и достоверность разработанной методики оценки надежности резервированной программно-аппаратной МАС, описанной в разделе 5.

Вычислительный эксперимент выполнен на мобильной персональной ЭВМ со следующими характеристиками: центральный процессор – Intel Core i7-2620M, объем оперативного запоминающего устройства – 8096Mb, операционная система – Microsoft Windows 7 Professional Service Pack 1. Была проведена оценка среднего времени, необходимого для:

- составления функции работоспособности тестовой МАС;
- ортогонализации функции работоспособности тестовой МАС;

- получения упрощенной функции надежности тестовой МАС по ее функции работоспособности, записанной в виде ортогональной дизъюнктивной нормальной формы;
- вычисления вероятности безотказной работы тестовой МАС на интервале времени *[0; t]* по функции работоспособности, записанной в виде ортогональной дизъюнктивной нормальной формы, то есть фактически по неупрощенной функции надежности;
- вычисления вероятности безотказной работы тестовой МАС на интервале времени *[0; t]* по упрощенной функции надежности (34);
- получения экспериментальных значений вероятности безотказной работы тестовой МАС при проведении 10000 испытаний.

Результаты проведенной оценки приведены в таблице 9.

Таблица 9 – Оценка средних затрат времени

Действие	**Среднее время, мс**	**Количество экспериментов**
Составление функции работоспособности	0,17222	50000
Ортогонализация функции работоспособности	147,96600	1000
Получение и упрощение функции надежности	445,53500	1000
Вычисление вероятности безотказной работы по неупрощенной функции надежности	1,13287	100000
Вычисление вероятности безотказной работы по упрощенной функции надежности	0,00156	100000
Получение экспериментального значения вероятности безотказной работы	10,23400	1000

Согласно таблице 9 составим функции для получения временных затрат, необходимых для получения вероятности безотказной работы тестовой МАС на N интервалах времени *[0; t]* при использовании различных подходов. Время, необходимое для вычисления вероятности

безотказной работы тестовой МАС на N интервалах времени по неупрощенной функции надежности, вычислим по следующей формуле:

$$TCLC(N) = (0{,}17222 + 147{,}966) + 1{,}13287N = 148{,}13822 + 1{,}13287N, \quad (35)$$

где N – количество интервалов времени, для которых производится вычисление вероятности безотказной работы тестовой МАС.

Время, необходимое для вычисления вероятности безотказной работы тестовой МАС на N интервалах времени по упрощенной функции надежности, вычислим по следующей формуле:

$$TCLCS(N) = (148{,}13822 + 445{,}535) + 0{,}00156N = 593{,}67322 + 0{,}00156N, \quad (36)$$

где N – количество интервалов времени, для которых производится вычисление вероятности безотказной работы тестовой МАС.

Время, необходимое для получения экспериментального значения вероятности безотказной работы тестовой МАС на N интервалах времени, вычислим по следующей формуле:

$$TSIM(N) = 10{,}234N, \quad (37)$$

где N – количество интервалов времени, для которых производится получение экспериментального значения вероятности безотказной работы тестовой МАС.

Зависимости временных затрат, необходимых для получения вероятности работоспособности тестовой МАС, от количества интервалов времени при использовании различных подходов в соответствии с функциями (35) – (37) приведены на рисунках 3 и 4.

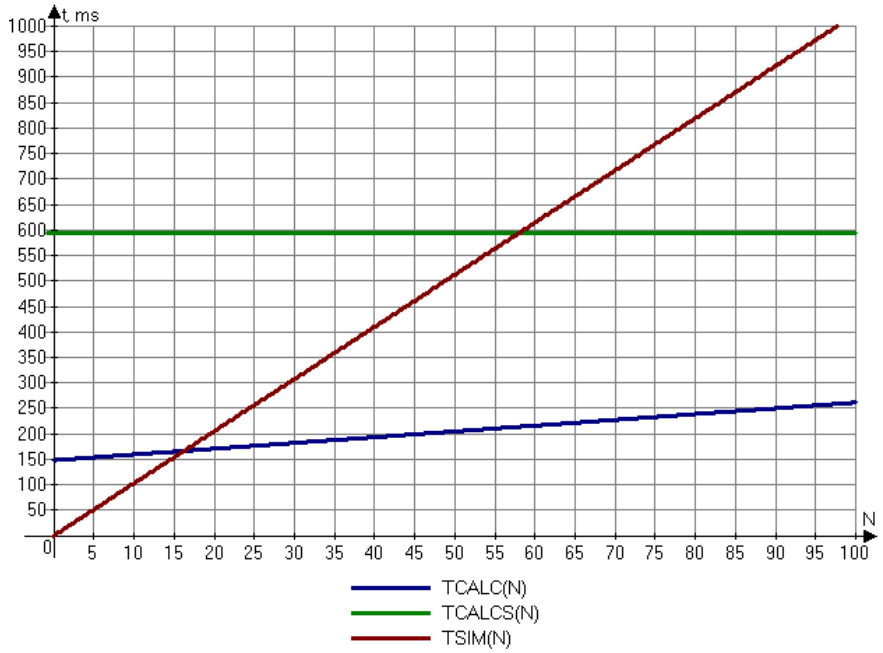


Рисунок 3 – Зависимость временных затрат от количества интервалов времени, для которых производится получение вероятности безотказной работы (до 100 интервалов)

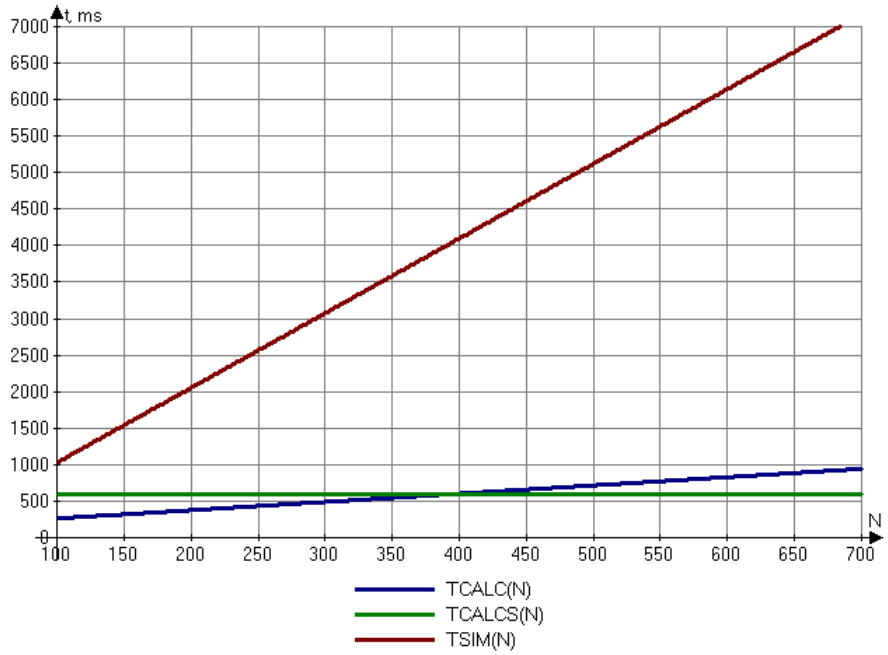


Рисунок 4 – Зависимость временных затрат от количества интервалов времени, для которых производится получение вероятности безотказной работы (до 700 интервалов)

Согласно формулам (35) – (37) при вычислении вероятности безотказной работы тестовой МАС по упрощенной функции надежности системы наблюдается наименьшее увеличение временных затрат при увеличении количества интервалов времени. Из графиков, приведенных на рисунках 3 и 4 следует, что при необходимости расчета вероятности безотказной работы на более чем 393 интервалах времени (или более чем 393 раза) наименьшие временные затраты соответствуют использованию упрощенной функции надежности (34). Однако при малых значениях количества интервалов времени вычислению вероятности безотказной работы тестовой МАС по упрощенной функции надежности (34) соответствуют наибольшие временные затраты, так как получение функции работоспособности МАС и упрощенной функции надежности требует значительных затрат времени. Отметим, что в разработанном комплексе программных средств, при помощи которого были проведены эксперименты, не реализованы методы оптимизации при решении задач ортогонализации дизъюнктивной нормальной формы и задач упрощения алгебраических выражений. Следовательно, начальные временные затраты, необходимые для вычисления вероятности безотказной работы МАС по упрощенной и неупрощенной функциям надежности, могут быть снижены.

Заключение

В статье представлена разработанная модель резервированной программно-аппаратной МАС, состоящей из задач, агентов, агентных платформ и исполнительных механизмов. Разработанная модель основана на использовании резервирования задач, как минимальных функциональных элементов МАС, и резервирования исполнительных механизмов, необходимых

для выполнения задач. Так как предложенная модель основана на предположении, что агентные платформы являются исполнительными контейнерами для агентов, а агенты в свою очередь являются исполнительными контейнерами для задач, то разработанная модель вводит избыточные множества таких контейнеров, не используя при этом резервирование агентов и агентных платформ исходной МАС. Кроме того, разработанная модель учитывает возможность объединения исполнительных механизмов МАС в конвейеры.

Для оценки надежности распределенных программно-аппаратных МАС были разработаны критерии работоспособности МАС, заданных предложенной авторами моделью. На основе полученных критериев работоспособности была разработана методика оценки надежности распределенных программно-аппаратных МАС с помощью логико-вероятностных методов [7], определяющая процедуру получения функции работоспособности для МАС, заданных описанной в статье моделью. Работоспособность и достоверность предложенного авторами подхода подтверждается проведенным вычислительным экспериментом, основанным на методе статистического моделирования. Кроме того, при проведении эксперимента были получены оценки временных затрат, необходимых для получения значений вероятности безотказной работы тестовой резервированной МАС.

Согласно предложенным критериям работоспособности для успешного функционирования МАС достаточно конфигурации системы, обеспечивающей возможность выполнения по крайней мере одной задачи каждого типа. Поэтому дальнейшие направления исследований в данной области должны быть связаны с разработкой методики восстановления работоспособности резервированных программно-аппаратных МАС, обеспечивающей выбор и выполнение задачи определенного типа из избыточного множества задач при отказах различных компонентов системы. Если конфигурация МАС не обеспечивает выполнение задачи определенного типа, то фактически отказ МАС происходит только в том случае и только в тот момент времени, когда задача данного типа должна быть выполнена. Поэтому после разработки методики восстановления работоспособности следует провести вычислительный эксперимент, основанный на статистическом моделировании отказов компонентов системы и на имитационном моделировании поведения МАС, то есть выполнения соответствующих задач как реакции на внешние или внутренние события и восстановления работоспособности при обнаружении отказов компонентов системы. Целью такого эксперимента является получение статистических вероятностей безотказной работы резервированной МАС, при этом отказ МАС должен фиксироваться в ситуации, когда задача определенного типа должна, но не может быть выполнена.

Разработанная методика оценки надежности резервированных МАС позволяет использовать преимущества логико-вероятностных методов, в том числе получение функции надежности в аналитической форме для расчета показателей надежности на стадии проектирования, при использовании многоагентного подхода к разработке систем промышленного уровня. Так как

разработанная модель резервированной распределенной программно-аппаратной МАС определяет структуру и конфигурацию такой МАС, а также задает стратегию обеспечения отказоустойчивости, то на основе данной модели могут быть сформулированы оптимизационные задачи синтеза резервированных МАС с заданной или максимальной надежностью, расчет которой позволяет провести предложенная в статье методика.

Список литературы

1. Jennings N.R. On agent-based software engineering // Artificial intelligence. 2000. Vol. 117, no. 2. P. 277-296.

2. Pullum L.L. Software fault tolerance techniques and implementation. Artech House, 2001. 360 p.

3. Cohen P.R., Kumar S. Towards a fault-tolerant multi-agent system architecture // Proceedings of the Fourth International Conference on Autonomous Agents. ACM, 2000. P. 459-466. DOI: 10.1145/336595.337570

4. Guessoum Z., Briot J.P., Faci N. Towards Fault-Tolerant Massively Multiagent Systems // In: Massively Multi-Agent Systems I. Springer Berlin Heidelberg, 2005. P. 55-69. (Ser. Lecture Notes in Computer Science; vol. 3446). DOI: 10.1007/11512073_5

5. Serugendo G.D.M., Romanovsky A. Designing Fault-Tolerant Mobile Systems // In: Scientific Engineering for Distributed Java Applications. Springer Berlin Heidelberg, 2003. P. 185-201. (Ser. Lecture Notes in Computer Science; vol. 2604). DOI: 10.1007/3-540-36520-6_17

6. Mellouli S. A Reorganization Strategy to Build Fault-Tolerant Multi-Agent Systems // In: Advances in Artificial Intelligence. Springer Berlin Heidelberg, 2007. P. 61-72. (Ser. Lecture Notes in Computer Science; vol. 4509). DOI: 10.1007/978-3-540-72665-4_6

7. Рябинин И.А. Логико-вероятностный анализ и его современные возможности // Моделирование и Анализ Безопасности и Риска в Сложных Системах: труды Международной Научной Школы МА БР - 2010 (Санкт-Петербург, 6 - 10 июля 2010 г.). СПб.: ГУАП, 2010. С. 19-27.

8. Wooldridge M., Jennings N.R. Intelligent agents: Theory and practice // Knowledge engineering review. 1995. Vol. 10, no. 2. P. 115-152.

9. Рябинин И.А., Черкесов Г.Н. Логико-вероятностные методы исследования надежности структурно-сложных систем. М.: Радио и связь, 1981. 264 с.

10. Черкесов Г.Н. Методы и модели оценки живучести сложных систем: Конспект лекций. М.: Знание, 1987. 55 с.

SCIENTIFIC PERIODICAL OF THE BAUMAN MSTU
SCIENCE and EDUCATION
EL № FS77 - 48211. №0421200025. ISSN 1994-0408
electronic scientific and technical journal

Reliability assessment for redundant multi-agent systems

 # 01, Januare 2014
DOI: 10.7463/0114.0696290

A.M. Igumnov, S.E. Saradgishvili

St. Petersburg State Polytechnical University, St. Petersburg, Russia
Alexei.Igumnov@gmail.com
Ssaradg@yandex.ru

The application of logical-and-probabilistic methods to evaluation of reliability indexes of distributed hardware-software multi-agent systems is examined in the article. The developed model of multi-agent system defines fault-tolerance technique, which is based on replication of tasks and actuators of the existing system as well as on introduction of redundant sets of executive containers. The developed model is also applicable for multi-agent systems with actuators being aggregated into a set of production lines. Criteria of serviceability and the methodology for assessment of reliability, which defines the procedure of construction of an analytic operability function, were developed on a basis of the introduced model. The developed methodology was validated by the computing experiment described in the article. The model of redundant multi-agent system and the introduced reliability assessment methodology could be applied to various problems of synthesis of optimal multi-agent system with required reliability indexes.

Publications with keywords: reliability, operating capability, redundancy, operability function
Publications with words: reliability, operating capability, redundancy, operability function

References

1. Jennings N.R. On agent-based software engineering. *Artificial intelligence*, 2000, vol. 117, no. 2, pp. 277-296.

2. Pullum L.L. *Software fault tolerance techniques and implementation*. Artech House, 2001. 360 p.

3. Kumar S., Cohen P.R. Towards a fault-tolerant multi-agent system architecture. In: *Proceedings of the Fourth International Conference on Autonomous Agents*. ACM, 2000, pp. 459-466. DOI: 10.1145/336595.337570

4. Guessoum Z., Briot J.P., Faci N. Towards Fault-Tolerant Massively Multiagent Systems. In: *Massively Multi-Agent Systems I*. Springer Berlin Heidelberg, 2005. P. 55-69. (Ser. *Lecture Notes in Computer Science*; vol. 3446). DOI: 10.1007/11512073_5

5. Serugendo G.D.M., Romanovsky A. Designing Fault-Tolerant Mobile Systems. In: *Scientific Engineering for Distributed Java Applications*. Springer Berlin Heidelberg, 2003. P. 185-201. (Ser. *Lecture Notes in Computer Science*; vol. 2604). DOI: 10.1007/3-540-36520-6_17

6. Mellouli S. A Reorganization Strategy to Build Fault-Tolerant Multi-Agent Systems. In: *Advances in Artificial Intelligence*. Springer Berlin Heidelberg, 2007. P. 61-72. (Ser. *Lecture Notes in Computer Science*; vol. 4509). DOI: 10.1007/978-3-540-72665-4_6

7. Ryabinin I.A. [Logic and probabilistic analysis and its modern capabilities]. *Modelirovanie i Analiz Bezopasnosti i Riska v Slozhnykh Sistemakh: trudy Mezhdunarodnoy Nauchnoy Shkoly MA BR - 2010* [Modeling and Analysis of Safety and Risk in Complex Systems: Proc. of the International Scientific School MA SR - 2010], St. Petersburg, 6 - 10 July 2010. St. Petersburg, Publ. of St. Petersburg State University of Aerospace Instrumentation, 2010, pp. 19-27. (in Russian)

8. Wooldridge M., Jennings N.R. Intelligent agents: Theory and practice. *Knowledge engineering review*, 1995, vol. 10, no. 2, pp. 115-152.

9. Ryabinin I.A., Cherkesov G.N. *Logiko-veroyatnostnye metody issledovaniya nadezhnosti strukturno-slozhnykh system* [The logic-probabilistic research methods of structure-complex systems reliability]. Moscow, Radio i svyaz' Publ., 1981. 264 p. (in Russian)

10. Cherkesov G.N. *Metody i modeli otsenki zhivuchesti slozhnykh sistem: Konspekt lektsiy* [Methods and models of complex systems survivability assessment: Lecture notes]. Moscow, Znanie Publ., 1987. 55 p. (in Russian)