

ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗАЦИИ ОЦЕНКИ РИСКА ОТКАЗОВ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

В.В. Вычужанин, Н.Д. Рудниченко, А.В. Вычужанин, А.Е. Козлов

Одесский национальный политехнический университет,
просп. Шевченко, 1, Одесса, 65044, Украина; e-mail: 126.ist.onpu@gmail.com

Надежность сложных технических систем зависит от многочисленных факторов, характеризующих их конструкцию, условия производства и эксплуатации. Проведенный анализ процессов изменения технического состояния сложных технических систем показал, что процессы носят случайный характер. Оценка и анализ показателей надежности систем нуждается в использовании методов теории вероятности. Перспективным в таких случаях является применение когнитивного имитационного моделирования для диагностики уровня безопасности сложных технических систем. Существующая теоретическая база и наличие широкого спектра программного обеспечения имитационного моделирования в виде таких продуктов, как Arena, AutoMod, AnyLogic, Extend, GPSS World и других способствует активному применению когнитивного имитационного моделирования для оценки риска отказов сложных технических систем. Однако известные программные средства облегчают только сам процесс испытаний и не решают трудоемкие задачи, связанные с получением исходной информации о техническом состоянии систем, а именно ее интерпретацию, формализацию и адекватное соотношение с конкретным объектом. На базе ранее разработанной концепции диагностики состояния сложных технических систем создана программная реализация обеспечения автоматизированного процесса оценки риска отказов, а также ущерба от отказов элементов и межэлементных связей сложных технических систем. Разработанное программное обеспечение автоматизации оценки риска отказов сложных технических систем при их диагностике базируется на клиент-серверной архитектуре и позволяет выявить наименее работоспособные элементы и межэлементные связи, функционирование которых существенно отражается на работоспособности и надежности всей системы.

Ключевые слова: сложная техническая система, риск отказов, когнитивная имитационная модель, программное приложение, диаграммы прецедентов, диаграммы классов использования программного обеспечения.

Введение

Надежность сложных технических систем (СТС) зависит от многочисленных факторов, характеризующих их конструкцию, условия производства и эксплуатации [1,2]. Надежность СТС зависит от риска отказов элементов систем и их подсистем. Поэтому решение задач, связанных с оценкой риска отказов СТС, способствует повышению надежности систем [3-6].

Процессы изменения технического состояния СТС носят случайный характер, а при оценке и анализе показателей надежности систем необходимо использовать методы теории вероятности [7,8].

Для диагностики риска отказов СТС в таких случаях перспективным является использование когнитивного имитационного моделирования (КИМ) [1,9-11], использующего модели в виде ориентированных графов, отражающих взаимодействие элементов СТС.

Существующая теоретическая база и наличие широкого спектра программного обеспечения имитационного моделирования в виде мощных продуктов, таких как

Arena, AutoMod, AnyLogic, Extend, GPSS World и других способствует активному применению когнитивного имитационного моделирования для оценки риска отказов СТС [12-15]. Однако известные программные средства облегчают только сам процесс испытаний и не касаются самой трудоемкой задачи сбора исходной информации, ее интерпретации, формализации и адекватного соотношения с конкретным объектом. Освоение таких программных сред требует значительных усилий.

На базе, описанной в [7] концепции оценки риска отказов элементов СТС, становится возможной разработка программного обеспечения, позволяющего осуществлять автоматизированный процесс оценки риска отказов элементов и межэлементных связей СТС с учетом их технического состояния [16-18].

Цель работы

Целью статьи является разработка кроссплатформенного программного обеспечения для автоматизации процессов оценки риска отказов СТС.

Программное обеспечение автоматизации оценки риска отказов СТС

Для определения общих границ и контекста моделируемой предметной области на начальных этапах разработки программного обеспечения автоматизации оценки риска отказов СТС, ущербов от потери работоспособности системы и формулирования общих требований к ее поведению используются диаграммы вариантов использования (прецедентов) (рис.1).

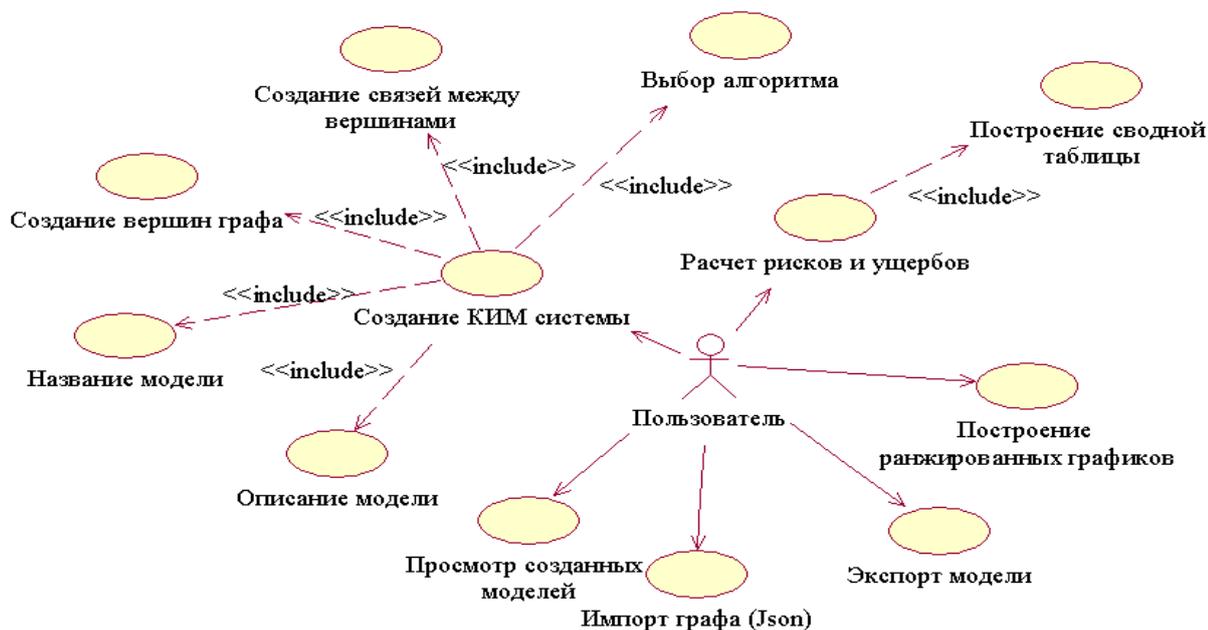


Рис. 1. Диаграмма вариантов использования программного обеспечения

Разработанное программное обеспечение позволяет пользователю:

- создавать КИМ СТС в виде графа, с поддержкой функций названия модели, задания краткого описания, создания новой вершины и ее изображения на панели, создания связи между вершинами, выбора алгоритма расположения и отображения структуры в графическом контейнере;
- просматривать структуру ранее созданной КИМ в виде графа с отображением общего числа вершин и ребер, поддерживая операцию ее загрузки в рабочее пространство программы;

- импортировать графа в формате *.json для его визуализации в системе;
- экспортировать КИМ в виде графа в отдельный графический файл формата *.jrg;
- проводить расчеты значений ущербов от отказов и риска отказов элементов моделируемой СТС и отображать полученные результаты в табличном виде сводной таблицы;
- построить графики визуализации полученных результатов в ранжированном виде.

Диаграмма вариантов использования программного обеспечения позволяет разработать модели основных сущностей реализации проекта путем создания диаграммы классов. Фрагмент созданной диаграммы классов программного обеспечения приведен на рисунке 2.

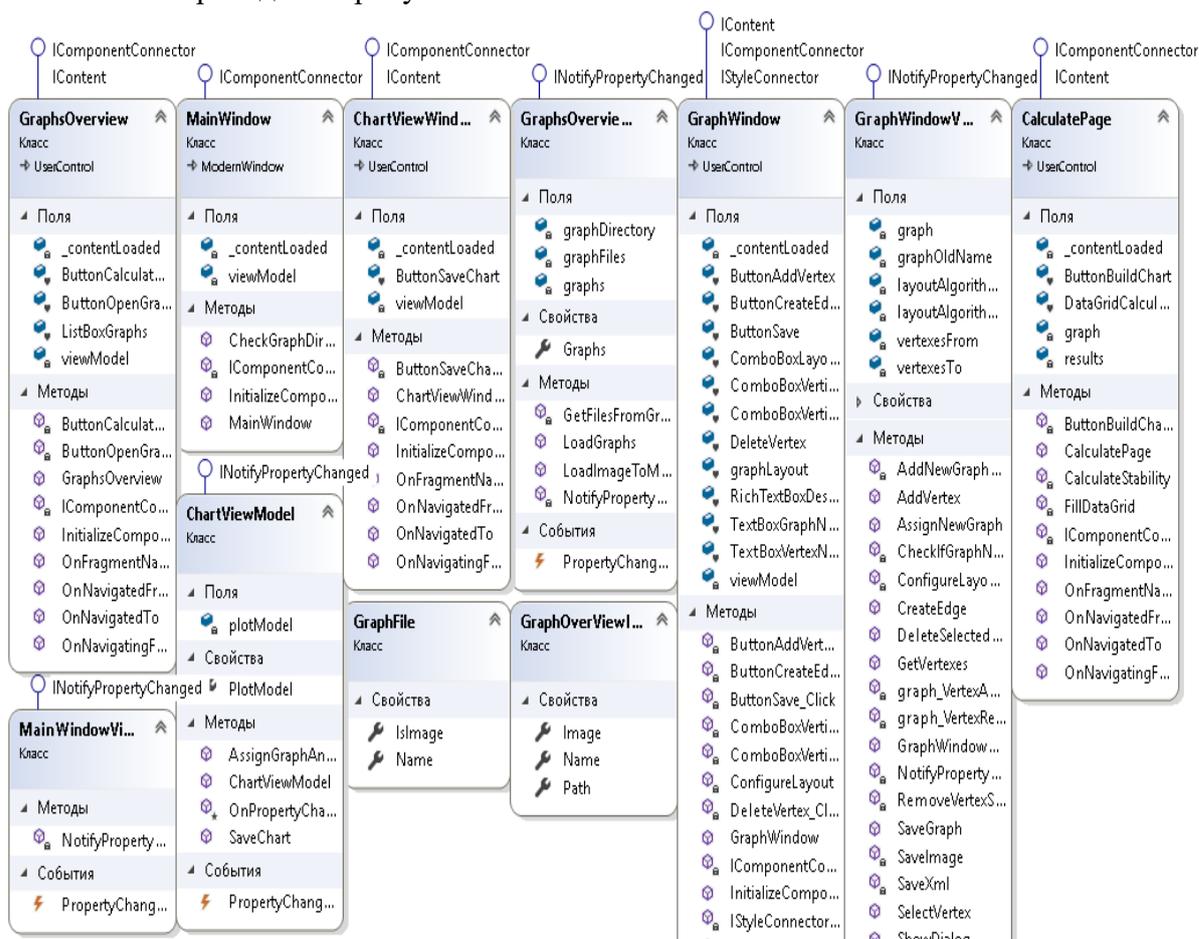


Рис. 2. Диаграмма классов программного обеспечения

Классы *GraphsOverview*, *MainWindow*, *ChartViewWindow*, *GraphsOverview Window*, *GraphWindow*, *GraphWindowView*, *CalculatePage*, *MainWindowView* и *ChartViewModel* имплементируют ряд интерфейсов для обеспечения необходимых функциональных возможностей по визуализации обрабатываемых и расчетных данных. Для реализации обозначенной логики использованы и задействованы следующие интерфейсы для КИМ в виде графа:

- *IComponentConnector* (для обеспечения связи объектов элементов между собой);
- *IContent* (для отображения и реализации функций динамической привязки и обрисовки объектов рабочего графической контейнера на соответствующей форме);

- INotifyPropertyChanged (для обработки событий изменения свойств созданных графических объектов);

- IStyleConnector (для изменения типа связи между элементами).

На основе созданных диаграмм описаны особенности физического представления системы в виде формализации порядка взаимосвязи базовых элементов системы. Для чего используется диаграмма компонентов (рис.3), позволяющая определять архитектуру системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Главный модуль (MainApp) осуществляет функции вызова соответствующих модулей для обработка пользовательских запросов по: построению модели графа (GraphBuilder), на базе использования внешних зависимостей Graph# и OxyPlot, а также модуля WPF GraphLayout для обеспечения работы интерактивного контейнера визуализации полученной модели; расчету значений ущерба и риска отказов модели системы; просмотру полученных результатов в табличном виде для оценки их значений; построению и просмотру графика, отображающего результаты расчетов в ранжированном виде.

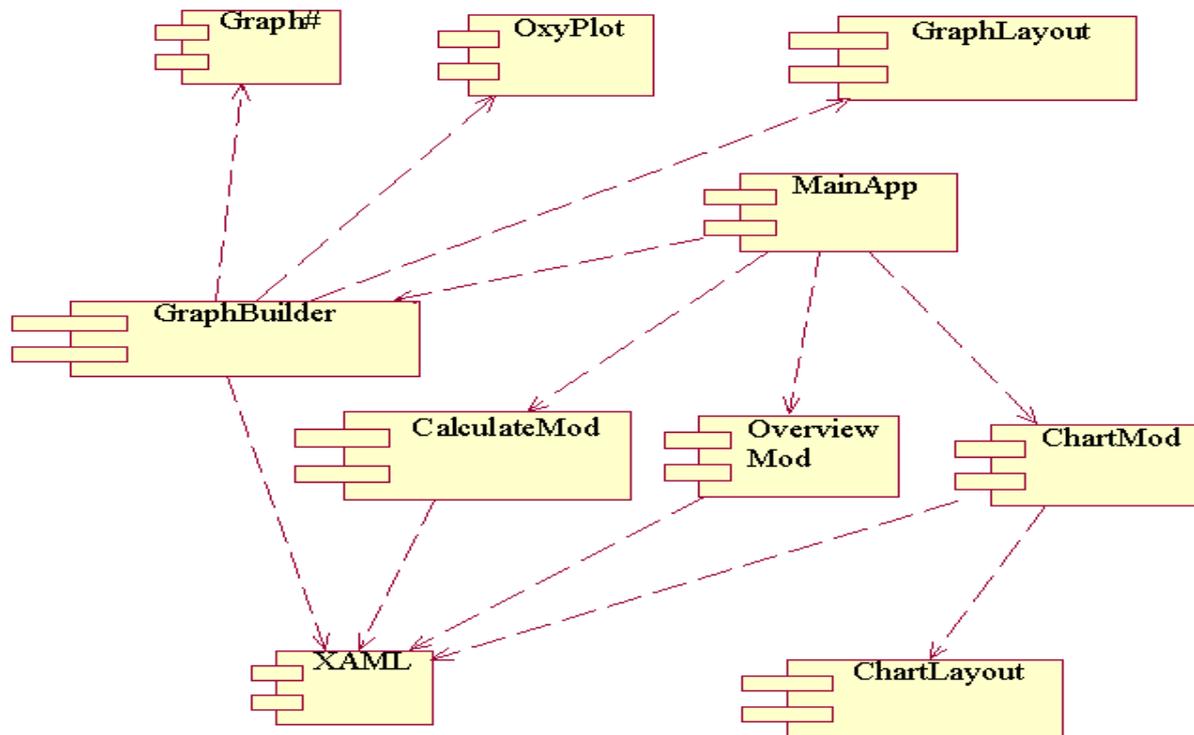


Рис. 3. Диаграмма компонентов программного обеспечения

В качестве графических библиотек, подключенных как внешние зависимости, использованы Graph# и OxyPlot. Graph# (библиотека для визуализации графов), содержащая некоторые алгоритмы компоновки, а также для контроля GraphLayout WPF приложений. Поддерживаемые алгоритмы компоновки и удаления: Fruchterman - Reingold; Kamada - Kawai; ISOM; LinLog; простой макет дерева; Sugiyama; Force-Scan; алгоритм выделения.

В целях моделирования взаимодействия объектов в проектируемом программном обеспечении во времени, а также обмена сообщениями между ними используется диаграмма последовательности (рис. 4), являющаяся одной из разновидности диаграмм взаимодействия.

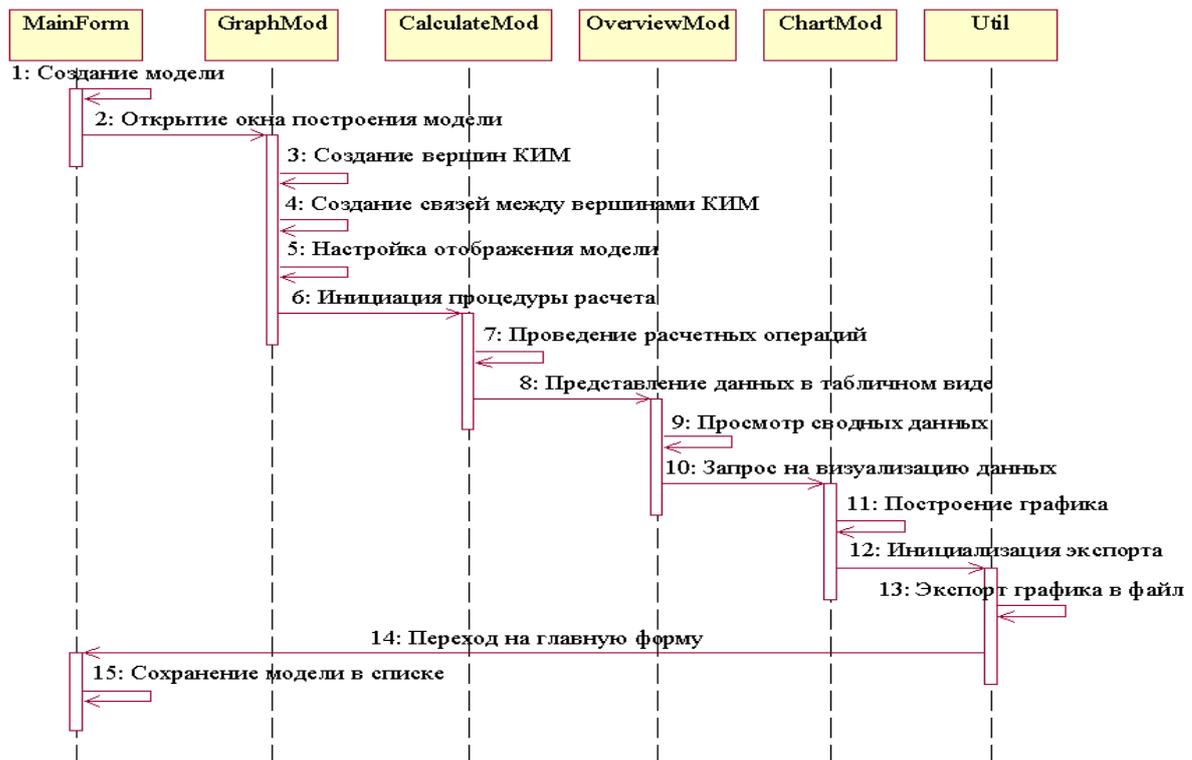


Рис. 4. Диаграмма последовательности действий программного обеспечения

Каждая из приведенных форм (кроме главной) является отдельным фрагментом, динамически загружаемым в едином пространстве главной формы в виде новой вкладки (объект `TabPage`). Базовой составляющей метода построения структуры графа является алгоритм Сугиямы. В его основе лежат основные этапы: распределение вершин по уровням так, чтобы ребра были одного направления и длина каждого из них была минимальной; минимизация количества фиктивных вершин, т.к. время выполнения следующих шагов зависит от суммарного числа вершин, а изгибы ребер в конечном изображении появляются только на местах фиктивных вершин и снижают наглядность изображения; минимизация количества пересечений ребер путем изменения порядка вершин на каждом уровне; выбор координат вершин на каждом уровне для минимизации количества изгибов ребер.

В целях описания программного функционала обозначим ключевые классы, реализующие бизнес-логику работы созданного программного обеспечения:

- класс проведения расчетов `public partial class CalculatePage: UserControl, IContent, IComponentConnector`; интерпретирует и использует полученные результаты построения модели системы в виде графа для оценки ущерба и риска отказов;
- класс построения и отображения ранжированного графика расчетных значений риска отказов `public class ChartViewModel: INotifyPropertyChanged`;
- класс построения КИМ в виде графа `class GraphWindowViewModel: INotifyPropertyChanged`;
- класс отображения полученной модели графа системы `public class GraphsOverview: UserControl, IContent, IComponentConnector`.

Интерфейс формы создания графа КИМ СТС разработанного программного обеспечения приведен на рисунке 5. В приведенном окне можно построить новый граф или редактировать существующий. Такая форма позволяет выбирать один из поддерживаемых алгоритмов построения и отображения графа в рамках контейнера, задать ему имя в виде строки, а также ввести его текстовое описание и сохранить созданную модель в формате `*.xml`.

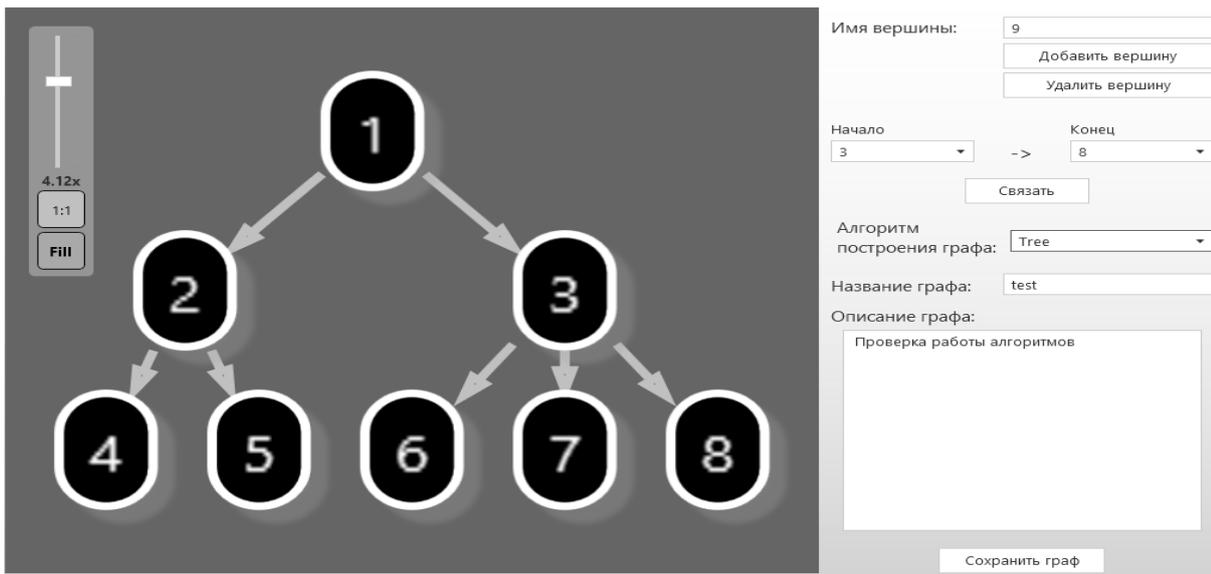


Рис. 5. Интерфейс формы создания графа КИМ СТС разработанного ПО

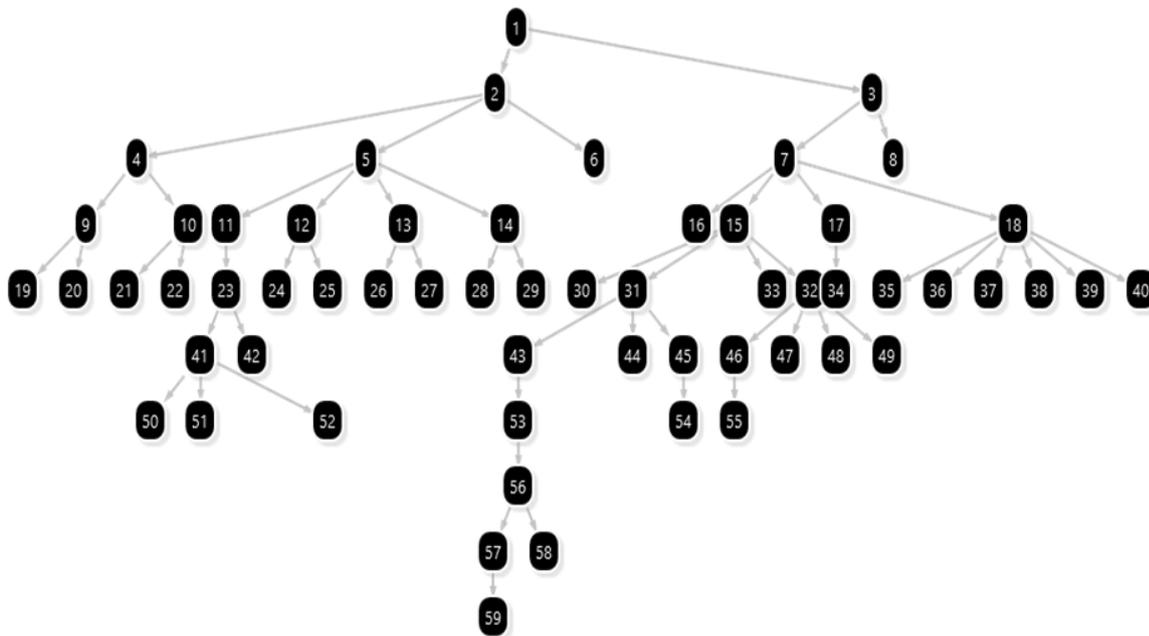


Рис. 6. КИМ СТС (1- рулевая машина, 2- сегмент червячного колеса и тормоза, 3-червь, 4- румпель, 5-редуктор, 6- рулевой штырь, 7 - рулевой сектор, 8-полуось, 9-кранштейн поддона, 10- болт, 11-болт с гайкой, 12- шайба, 13-стопорная планка, 14, 15, 16, 24, 25-шестерни, 17-водило, 18-свободный эпицикл, 19-зубчатые колеса, 20-свободное водило, 21, 22-валы, 23-тормозной эпицикл, 26-двигатель, 27-пружина, 28-баллер руля, 29-профильный руль, 30-приводное колесо, 31-гребной вал, 32, 33-валы турбин низкого и высокого давления, 34-наддувочный агрегат, 35-приводное колесо, 36-промежуточные шестерни, 37-приводное колесо коленчатого вала, 38-распределительный вал, 39-шатун, 40-поршень, 41-цилиндровая гильза, 42-камера охлаждающей воды, 43-коленчатый вал, 44-охладитель наддувочного воздуха, 45-трубопровод отработавших газов, 46,47-трубопроводы наддувочного воздуха и охлаждающей воды, 48, 49- масляный и топливный трубопроводы, 50-штанга, 51-топливный насос, 52-масляное кольцо, 53-крышка цилиндра 54, 55, 56-выпускной, впускной и топливный клапаны, 57-станина двигателя, 58-масляная ванна картера, 59-блок цилиндров

Для апробации программного приложения создан в ней КИМ СТС в виде ориентированного графа на примере векторного управления рулевой передачей с электрическим приводом морского судна (рис. 6) [19,20]. По результатам проведенных расчетов формируется соответствующая таблица, а также график (рис.7) оценки риска отказов элементов системы, на основе полученных при расчете данных. Поддерживаются функции фильтрации данных по колонкам таблицы в удобном виде. Из проведенных расчетов следует, что наиболее уязвимыми элементами системы являются рулевой сектор, червь, сегмент червячного колеса и тормоза, редуктор. Отказ их представляют наибольшую угрозу для функционирования всей системы векторного управления рулевой передачей с электрическим приводом. В этой связи необходимо осуществлять постоянный мониторинг фактического состояния и обеспечить их резервирование. Это позволит избежать потери работоспособности элементов системы, снизить риск их отказов.

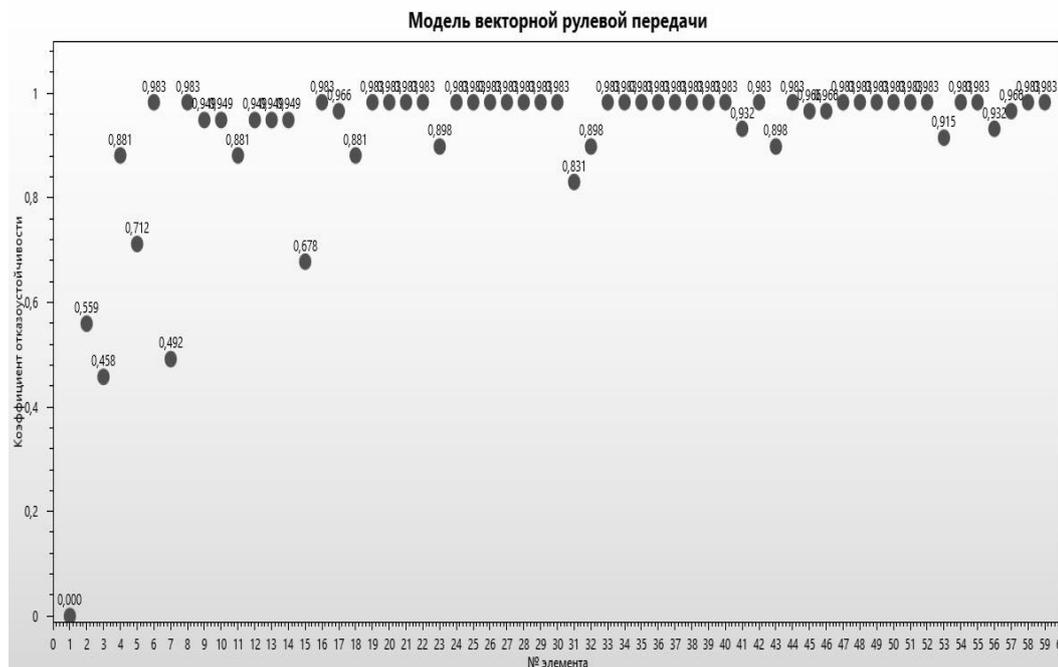


Рис. 7. График оценки рисков элементов системы

Выводы

Разработанное программное обеспечение автоматизации оценки риска отказов сложных технических систем при их диагностике систем базируется на клиент-серверной архитектуре.

Применение разработанного программного обеспечения автоматизации оценки риска отказов сложных технических систем позволяет выявить наименее работоспособные элементы и межэлементные связи, функционирование которых существенно отражается на работоспособности и надежности всей системы.

Список литературы

1. Вычужанин В.В., Рудниченко Н.Д. Методы информационных технологий в диагностике состояния сложных технических систем: монография. Одесса: Экология, 2019. 178 с.
2. Цветков В.Я. Сложные технические системы. *Образовательные ресурсы и технологии*. 2017. №3(20). С. 86-92.

3. Vychuzhanin V.V., Rudnichenko N.D. Assessment of risks structurally and functionally complex technical systems. *Eastern-European Journal of Enterprise Technologies*. 2014. V.1, №2. С. 18-22.
4. Вычужанин В.В., Рудниченко Н.Д. Технические риски сложных комплексов функционально взаимосвязанных структурных компонентов судовых энергетических установок. *Вісник Одеського національного морського університету*. 2014. Випуск 2(40). С. 68-77.
5. Вычужанин В.В., Рудниченко Н.Д. Метод управления рисками судовых сложных технических систем. *Проблеми техніки*. 2014. №2. С. 138-142.
6. Вычужанин В.В. Модель оценки живучести судовых технических систем. *Вестник Николаївського кораблебудівного університету*. 2012. №3. С. 62-67.
7. Вычужанин В.В., Рудниченко Н.Д., Вычужанин А.В., Козлов А.Е. Информационное обеспечение системы выявления риска отказов автомобильного оборудования. *Інформатика та математичні методи в моделюванні*. 2019. Т.9, №3. С. 121-133.
8. Дорохов А.Н., Керножицкий В.А., Миронов А.Н., Шестопалова О.Л. Обеспечение надежности сложных технических систем. СПб.: Издательство «Лань», 2011. 352 с.
9. Шеннон Р.Дж. Имитационное моделирование систем – искусство и наука. М.: Мир, 1978. 420 с.
10. Beetz M., Buss M., Wollherr D. Cognitive technical systems – what is the role of artificial intelligence. *Advances in Artificial Intelligence*. 2007. Pp. 19-42.
11. Рудниченко Н.Д., Вычужанин В.В. Информационная когнитивная модель технологической взаимозависимости сложных технических систем. *Інформатика та математичні методи в моделюванні*. 2013. Т.3, №3. С. 240-247.
12. Боев В.Д. Исследование адекватности GPSS World и AnyLogic при моделировании дискретно-событийных процессов: монография. СПб.: ВАС, 2011. 404 с.
13. Вивек Кале Внедрение SAP R/3: Руководство для менеджеров и инженеров: Пер. с англ. М.: Компания АйТи, 2006. 511 с.
14. Карпов Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic. СПб.: БХВ-Петербург, 2005. 400 с.
15. Маклаков С.В. Моделирование бизнес-процессов с AllFusion Process Modeler. М.: ДИАЛОГ МИФИ, 2003. 240 с.
16. Klein J.H., Cork R.B. An approach to technical risk assessment. *International Journal of Project Management*. 1998. Т.6, №16. Pp. 345-351.
17. Kertzner P., Watters J. Process Control System Security Technical Risk Assessment Methodology & Technical Implementation. *Research Report*. 2008. №13. 47 p.
18. O'Neill John, Nitin Thakur, Alan Duus Technical Risk Assessment: a Practitioner's Guide. Australia, 2007. 29 p.
19. Вычужанин В.В. Информационное обеспечение мониторинга и диагностирования технического состояния судовых энергоустановок. *Вісник Одеського національного морського університету, збірник наукових праць*. 2012. №35. С. 111-124.
20. Vychuzhanin V., Rudnichenko N., Boyko V., Shibaeva N., Konovalov S. Devising a method for the estimation and prediction of technical condition of ship complex systems. *Eastern-European Journal of Enterprise Technologies*. 2016. V.84, № 6/9. Pp. 4-11.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗАЦІЇ ОЦІНКИ РИЗИКУ ВІДМОВ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ

В.В. Вычужанин, Н.Д. Рудниченко, А.В. Вычужанин, А.Е. Козлов

Одесский национальный политехнический университет

просп. Шевченка, 1, Одеса, 65044, Україна; e-mail: 126.ist.onpu@gmail.com

Надійність складних технічних систем залежить від численних факторів, що характеризують їх конструкцію, умови виробництва і експлуатації. Проведений аналіз процесів зміни технічного стану складних технічних систем показав, що процеси носять випадковий характер. Оцінка і аналіз показників надійності систем потребує використання методів теорії ймовірності. Перспективним в таких випадках є застосування когнітивного імітаційного моделювання для діагностики рівня безпеки складних технічних систем. Існуюча теоретична база і наявність широкого спектра програмного забезпечення імітаційного моделювання у вигляді таких продуктів, як Arena, AutoMod, AnyLogic, Extend, GPSS World та інших сприяє

активному застосуванню когнітивного імітаційного моделювання для оцінки ризику відмов складних технічних систем. Однак відомі програмні засоби полегшують тільки сам процес випробувань і не вирішують трудомісткі завдання, пов'язані з отриманням вихідної інформації про технічний стан систем, а саме її інтерпретацію, формалізацію і адекватне співвідношення з конкретним об'єктом. На базі раніше розробленої концепції діагностики стану складних технічних систем створена програмна реалізація забезпечення автоматизованого процесу оцінки ризику відмов, а також шкоди від відмов елементів і межелементних зв'язків складних технічних систем. Розроблене програмне забезпечення автоматизації оцінки ризику відмов складних технічних систем при діагностиці систем базується на клієнт-серверній архітектурі і дозволяє виявити найменш працездатні елементи і межелементні зв'язки, функціонування яких істотно відбивається на працездатності і надійності всієї системи.

Ключові слова: складна технічна система, ризик відмов, когнітивна імітаційна модель, програмний додаток, діаграми прецедентів, діаграми класів використання програмного забезпечення.

EVALUATION AUTOMATION SOFTWARE RISK OF FAILURES OF COMPLEX TECHNICAL SYSTEMS

V.V. Vyuzhujanin, N.D. Rudnichenko, A.V. Vyuzhujanin, A.E. Kozlov

Odessa National Polytechnic University,
Shevchenko Ave., 1, Odessa, 65044, Ukraine; e-mail: 126.ist.onpu@gmail.com

The reliability of complex technical systems depends on numerous factors characterizing their design, production and operating conditions. The analysis of the processes of changing the technical condition of complex technical systems showed that the processes are random in nature. Evaluation and analysis of reliability indicators of systems requires the use of methods of probability theory. Promising in such cases is the use of cognitive simulation to diagnose the level of security of complex technical systems. The existing theoretical base and the presence of a wide range of simulation software in the form of products such as Arena, AutoMod, AnyLogic, Extend, GPSS World and others contribute to the active use of cognitive simulation to assess the risk of failure of complex technical systems. However, well-known software tools facilitate only the testing process itself and do not solve the laborious tasks associated with obtaining initial information about the technical condition of the systems, namely its interpretation, formalization and adequate relationship with a specific object. On the basis of the previously developed concept for diagnosing the state of complex technical systems, a software implementation has been created for providing an automated process for assessing the risk of failures, as well as damage from failures of elements and inter-element connections of complex technical systems. The developed software for automating the risk assessment of failures of complex technical systems during system diagnostics is based on the client-server architecture and allows to identify the least efficient elements and interconnections, the functioning of which significantly affects the operability and reliability of the entire system.

Keywords: complex technical system, failure risk, cognitive simulation model, software application, use case diagrams, software usage class diagrams.