

# РАЗРАБОТКА ИНСТРУМЕНТАЛЬНОГО СРЕДСТВА ВИЗУАЛЬНОГО ПРОЕКТИРОВАНИЯ И ВАЛИДАЦИИ БАЗОВЫХ ФУНКЦИОНАЛЬНЫХ БЛОКОВ СТАНДАРТА IEC 61499

Зорин В.Н., аспирант,

Пензенский Государственный Университет, г. Пенза, Россия

**Аннотация.** В статье рассматривается процесс разработки инструментального средства для визуального проектирования базовых функциональных блоков стандарта IEC 61499. Ключевыми компонентами средства являются: графический редактор диаграммы управления состояниями, средства импорта и экспорта модели, модуль валидации графа состояний и инструмент преобразования в формат .fbt. Инструментальное средство позволяет сохранять целостность модели при импорте или экспорте, а также обнаруживает структурные нарушения с использованием инструмента валидации.

**Ключевые слова:** базовый функциональный блок, IEC 61499, FBDK, 4DIAC, инструментальное средство, киберфизические системы.

В настоящее время промышленное производство характеризуется переходом к концепции Индустрия 4.0. Наблюдается изменения автоматизации производства в сторону более сетевых устройств управления, которые требуют новые парадигмы и языки программирования [5]. Основным требованием в её рамках является способность быстрой адаптации к изменениям производственных задач без существенных изменений программного обеспечения. В последние годы требования к совместимости, гибкости и возможности реконфигурации сложных приложений в сфере автоматизации значительно возросли [2]. В современных индустриальных системах, распределённые автоматизированные системы играют решающую роль в обеспечении гибкости, масштабируемости и надёжности [1]. Классические подходы к программированию промышленных систем, основанных на

стандарте IEC 61131-3, используют последовательное выполнение команд и опрос входных сигналов, однако монолитная структура кода программ усложняет повторное использование её компонентов. Альтернативой является стандарт IEC 61499, построенный на событийно-ориентированной архитектуре и обеспечивающий детерминизм управления системой и распределения логики между устройствами. Она ориентирована на проектирование и реализацию распределённого программного обеспечения для промышленной автоматизации аппаратно-независимым способом [4]. Также согласно данному стандарту, эталонная архитектура системы включает в себя один из нескольких независимых физических устройств, которые обмениваются данными через сеть [3].

Цель работы – разработать инструментальное средство визуального проектирования базовых функциональных блоков стандарта IEC 61499 со встроенной верификацией целостности структуры.

Научная новизна состоит в предложении алгоритма, позволяющего обеспечивать совместимость между средами FBDK и 4DIAC, а также предлагающего встроенную проверку структурных ошибок на всех этапах проектирования - от создания блока до формирования файла конфигурации.

Разработанное приложение построено с применением принципов объектно-ориентированного программирования на платформе .NET 8 с использованием WinForms для создания пользовательского интерфейса. Архитектура приложения построена по принципу разделения модели конфигурации блока и интерфейса ввода данных, где модель функционального блока взаимодействует с механизмами верификации и модулем экспорта в xml-формат для различных сред (4DIAC или FBDK). Пользовательский интерфейс, показанный на рисунке 1, предоставляет возможность заполнения основных параметров модели, включая порты и алгоритмы, а также визуализирует модель графа ЕСС-состояний.

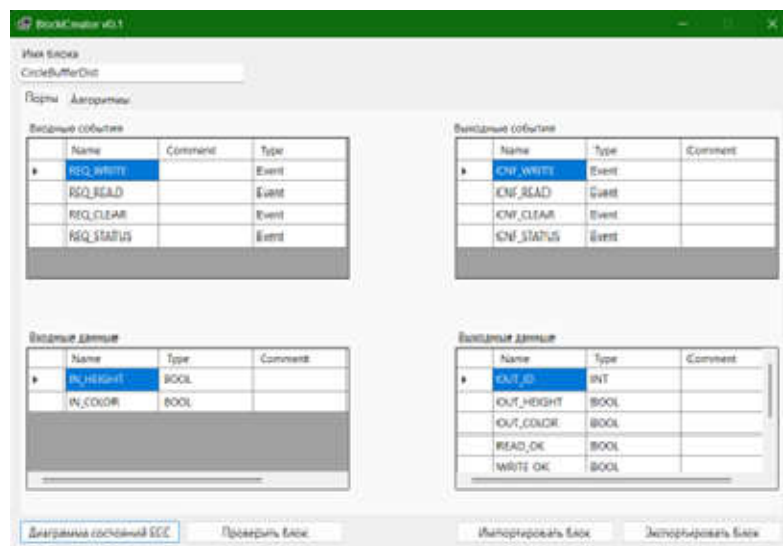


Рисунок 1 – Стартовое окно приложения

Интерфейс приложения разделён на следующие контексты:

- редактор портов и алгоритмов ST;
- редактор диаграмм ECC, позволяющий проектировать состояния и переходы внутри блока. Он представлен на рисунке 2.

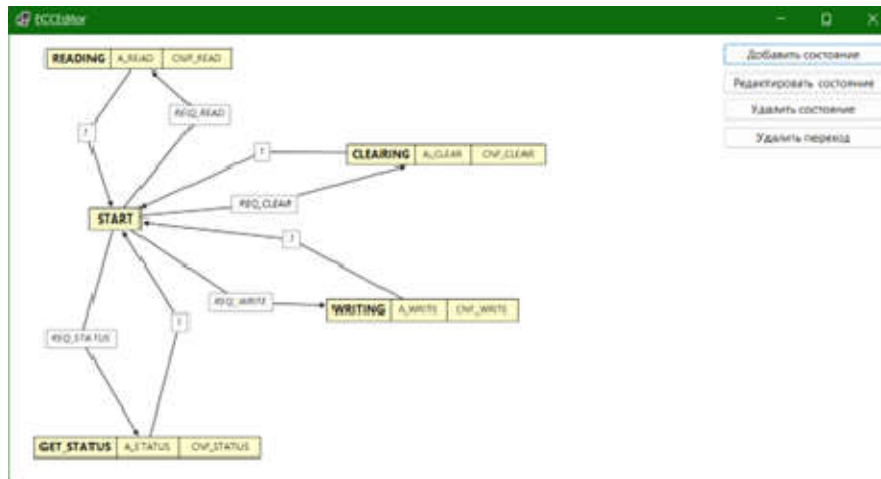


Рисунок 2 – Пример использования редактора диаграмм ECC

Применение библиотеки GDI+, при проектировании, позволяет упростить адаптацию приложения к различным операционным системам.

Рассмотрим интерфейс более детально. Диаграмма состояний формализуется, как ориентированный граф, состоящий из вершин и дуг, которые соответствуют состояниям и переходам. Состояние может быть

ассоциировано с алгоритмом и выходным событием. Редактор поддерживает интерактивное изменение топологии графа с динамическим перерасчётом связей и актуализацией изменений в модели.

Данные модели функционального блока формируются в формате .fbt, который является xml-документом, соответствующий спецификации стандарта IEC 61499. Корневой элемент FBType включает в себя метаданные (Identification, VersionInfo), описание интерфейса (InterfaceList), где указываются порты событий (EventInputs и EventOutputs) и порты данных (InputVars и OutputVars), и реализацию самого блока (BasicFB). Реализация включает в себя наличие графа состояний ECC (тег ECC, где ECState описывает состояние блока, ECAction – действие в рамках выбранного состояния, ECTransition – переходы между ними) и алгоритмов (ST).

Центральным классом средства является FuncBlockModel, описывающий данную структуру согласно стандарту IEC 61499:

```
public class FuncBlockModel
{
    public string Name { get; set; }

    public string Standard { get; set; }
    public string Comment { get; set; }
    public string Organization { get; set; }
    public string Version { get; set; }
    public string Author { get; set; }
    public string Date { get; set; }

    public string CompilerHeader { get; set; }
    public List<EventModel> EventInputs { get; set; } = new
List<EventModel>();
    public List<EventModel> EventOutputs { get; set; } = new
List<EventModel>();
    public List<DataModel> InputVars { get; set; } = new
List<DataModel>();
    public List<DataModel> OutputVars { get; set; } = new
List<DataModel>();
    public ECCStateModel ECCDiagram { get; set; }
    public List<AlgorithmModel> STAlgs { get; set; } = new
List<AlgorithmModel>();
}
```

Он гарантирует атомарность изменений и предотвращает рассинхронизацию данных между структурами блока. В случае, когда пользователь допускает ошибку при проектировании, средство выдаёт подробное описание ошибки, основанное на результате работы модуля валидации.

Модуль валидации предусматривает проверку целостности функционального блока по следующим критериям:

- уникальность идентификаторов в блоке;
- связность графа ЕСС;
- корректность привязки алгоритмов к состояниям в ЕСС;
- структурная валидность блока согласно стандарту IEC 61499.

Допустим, что в функциональном блоке присутствует тупиковое состояние, показанное на рисунке 3.

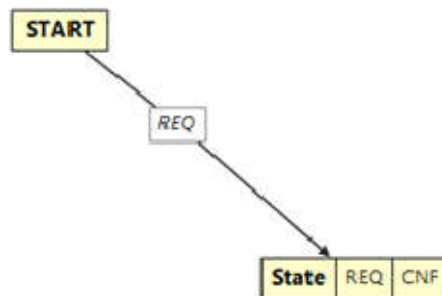


Рисунок 3 – ЕСС-диаграмма с тупиковым состоянием

Тогда в ходе валидации средство выведет предупреждение о наличии такого состояния. Оно показано на рисунке 4.

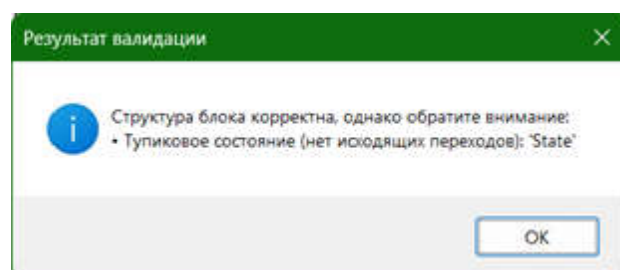


Рисунок 4 – Предупреждение о наличии тупикового состояния

В случае, если блок успешно проходит валидацию, то средство покажет сообщение о корректности структуры данного блока, показанное на рисунке 5.

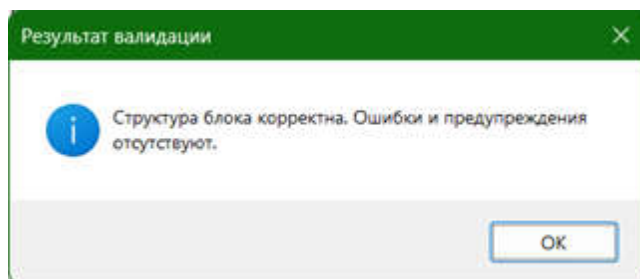


Рисунок 5 – Сообщение о корректности структуры функционального блока

Стоит отметить, что средство ограничено верификацией структурных и топологических уровней функционального блока. В данной реализации отсутствует семантический анализ кода на языке программирования ST.

Тем не менее, универсальность инструментального средства обеспечена строгим соответствием создаваемого файла по стандарту спецификации IEC 61499 и Holobloc. Основными механизмами являются маппинг модели функционального блока, сериализация основных атрибутов, нормализация заголовков и парсинг с безопасной навигацией по элементам.

Разработанное инструментальное средство обеспечивает верификацию топологии и связей базового функционального блока. Оно позволяет проверять корректность связей в графах состояний ЕСС, а также проверяет структурную валидность блока. Продолжается работа по совершенствованию программы с целью максимально доступного инструмента для проектирования функциональных блоков стандарта IEC 61499.

## Литература

1. Christensen J. H. et al. The IEC 61499 function block standard: Overview of the second edition //ISA autom. Week. – 2012. – Т. 6. – С. 6-7.

2. Dai W. et al. Bridging service-oriented architecture and IEC 61499 for flexibility and interoperability //IEEE Transactions on Industrial Informatics. – 2015. – T. 11. – №. 3. – C. 771-781.
3. Dubinin V., Vyatkin V., Pfeiffer T. Engineering of validatable automation systems based on an extension of UML combined with function blocks of IEC 61499 //Proceedings of the 2005 IEEE International Conference on Robotics and Automation. – IEEE, 2005. – C. 3996-4001.
4. Mousavi A., Vyatkin V. Energy Efficient Agent Function Block: A semantic agent approach to IEC 61499 function blocks in energy efficient building automation systems //Automation in construction. – 2015. – T. 54. – C. 127-142.
5. Prenzel L., Zoitl A., Provost J. IEC 61499 runtime environments: A state of the art comparison //International Conference on Computer Aided Systems Theory. – Cham : Springer International Publishing, 2019. – C. 453-460.