ФУНКЦИОНАЛЬНАЯ БЕЗОПАСНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ ВАЖНЫХ ДЛЯ БЕЗОПАСНОСТИ АЭС

Жарко Е.Ф.
Институт проблем управления им. В. А. Трапезникова РАН
zharko@ipu.ru

Аннотация: Проектирование систем важных для безопасности АЭС основано на выполнении требований нормативных документов и обеспечении качества программно-технических комплексов на протяжении всего жизненного цикла. Одной из важнейших задач в обеспечении качества программно-технических комплексов является процедура формирования требований к разрабатываемой или модифицируемой системе и последующая их верификация. Увеличение требований к качеству программного обеспечения для систем важных для безопасности АЭС на всех этапах жизненного цикла связано с возрастанием сложности и функциональности программного обеспечения и привело к необходимости разработки подходов для обоснования как безопасности самих систем, так и входящего в их состав программного обеспечения. В статье рассматривается подход, основанный на построении "функций безопасности" выполнение которых в дальнейшем верифицируется. Данный подход используется при верификации программного обеспечения для систем верхнего уровня АСУ ТП и может быть применен для анализа отказоустойчивости, информационной и кибер- безопасности программно-технических комплексов.

Ключевые слова: функциональная безопасность, программное обеспечение, функция безопасности, обеспечение качества, верификация, валидация, АЭС.

Введение

Информационные технологии играют ключевую роль в развертывании, эксплуатации и техническом обслуживании критически важных инфраструктур, которые имеют высокие требования к надежности и безопасности. Сбои (отказы) или аварии в системах объектов с повышенным риском эксплуатации, относящихся к критически важным инфраструктурам, могут:

- привести к разрушению или серьезным повреждениям дорогостоящего оборудования;
- нанести существенный ущерб окружающей среде;
- привести к угрозам здоровью и жизни людей.

Развитие автоматизации объектов критической инфраструктуры с повышенным риском эксплуатации, в том числе и в атомной энергетике, характеризуется тенденцией разработки автоматизированных систем управления технологическими процессами (АСУ ТП), реализующих значительно более сложные алгоритмы управления и анализа данных с использованием сложных программно-технических комплексов (ПТК) [1-4]. Разработка ПТК, их верификация и валидация, а также последующая их эксплуатация, а стечением времени, и модернизация, должны соответствовать и удовлетворять принятому уровню безопасности.

С возрастанием требований к объектам критической инфраструктуры, сложность программного обеспечения и его важность в обеспечении функций всей системы резко возрастает. Программное обеспечение (ПО) играет все более важную роль в выявлении и контроля опасных

факторов, а также в критических по отношению к безопасности функциях [5-9]. Широкое распространение программно-технических систем для объектов с повышенным риском эксплуатации привело к необходимости разработки методов для обоснования безопасности таких систем.

При обосновании безопасности в существующих подходах [10-14] использование моделей качества и безопасности играет центральную роль. В то время как системный подход к определению этих моделей по-прежнему редкость. Обеспечение качества программного обеспечения АСУ ТП на всех этапах его жизненного цикла базируется на качественном и количественном анализе, который, согласно нормативной документации, также должен проводиться на всех этапах. Качественный и количественный анализ качества программного обеспечения должен учитывать две составляющие программно-технических комплексов: аппаратную и программную [15].

ПО ПТК является неотъемлемым компонентом системы, влияющим на безопасность в целом, но при этом отсутствуют единые, универсальные и общепризнанные методы доказательства безопасности ПО. В связи с этим распространен подход, заключающийся в комплексном применении методов и средств повышения уровня безопасности системы на всех этапах жизненного цикла системы, а разработка новых методов верификации систем является актуальной задачей.

Выбор и определение функций безопасности относится к этапу валидации, выполняя задачу формализации задачи доказательства безопасности, и непосредственно влияет на качество последующей верификации ПТК.

1 Безопасность и программное обеспечение

Одной из актуальных проблем является обеспечение надежности функционирования систем важных для обеспечения безопасности АЭС. Требования, предъявляемые к безопасности таких систем, устанавливаются рядом международных стандартов. На их основе с учетом особенностей конкретного проекта формируется технология обеспечения безопасности ПТК на протяжении всего жизненного цикла. Существенной составляющей такой технологии является технология разработки программного обеспечения, обеспечивающая безопасность функционирования программного обеспечения:

- функциональную,
- технологическую,
- эксплуатационную.

Эти аспекты безопасности характеризуются следующим образом:

1) функциональная безопасность программного обеспечения состоит в способности противостоять непреднамеренным дестабилизирующим факторам, основными из которых являются отказы оборудования, технических средств ПТК и средств телекоммуникации, искажение исходной информации, дефекты и ошибки программного обеспечения, недостатки средств обнаружения опасных отказов;
2) эксплуатационная безопасность программного обеспечения состоит в его способности обеспечить безопасность информации в процессе штатной эксплуатации ПТК;
3) технологическая безопасность программного обеспечения состоит в его способности предотвращать в процессе функционирования ПТК воздействия, осуществляемые в целях несанкционированного раскрытия, изменения или уничтожения информации.

Между этими тремя аспектами имеются глубокие логические связи (например, на рушение технологической безопасности программного обеспечения может повлечь за собой нарушение его функциональной или эксплуатационной безопасности).

Программное обеспечение вносит существенный вклад в функции, выполняемые системами важными для безопасности. Программное обеспечение может поддерживать дополнительные функции, введенные в соответствии с проектом разрабатываемой или уже функционирующей системы. Для систем важных для безопасности АЭС жизненный цикл безопасности программного обеспечения тесно связан с жизненным циклом безопасности самой системы. Спецификация требований к программному обеспечению является частью спецификации системы. Качество ПО достигается благодаря применению методологии разработки и использованию методов верификации и валидации в течение жизненного цикла разработки ПО для систем важных для безопасности АЭС. Разработанный метод комплексной верификации программного обеспечения [16-18] основан на учете требований стандартов по безопасности, интегрирует этапы верификации ПО и их атрибуты, включая задействованный персонал, задачи, методики, устранение недостатков и выпускаемую документацию. Эффективность метода комплексной верификации программного

обеспечения была подтверждена в ходе работ по разработке информационных и управляющих систем, важных для безопасности АЭС.

ПО ПТК является неотъемлемым компонентом системы, влияющим на безопасность в целом, но при этом отсутствуют единые, универсальные и общепризнанные методы доказательства безопасности ПО. В связи с этим распространен подход, заключающийся в комплексном применении методов и средств повышения уровня безопасности системы на всех этапах жизненного цикла системы, а разработка новых методов верификации систем является актуальной задачей.

Одним из возможных способов поиска ошибок и повышения качества программного обеспечения, одним из аспектов которого является безопасность, является доказательство корректности, которое относится к формальным методам [19, 20]. Формальные методы в качестве доказательства корректности могут быть применены как для готового программного обеспечения, так и на ранних этапах разработки всего ПТК, но в любом случае одним из первых шагов верификации является определение функции безопасности, подлежащей проверке на корректность.

Доказательство безопасности разделяется на этапы валидации и верификации. На первом происходит определение функции безопасности (ФБ), а на втором проверяется, что она в рассматриваемом ПТК всегда выполняется. В нашем подходе данные этапы считаются независимыми, и при этом сначала проводится валидация, а в дальнейшем верификация, которая выполняется на основании полученной на предыдущем этапе ФБ. Анализ на безопасность необходимо выполнять в строгой последовательности.

2 Функция безопасности

Формальные методы в качестве доказательства корректности могут быть применены как для готового программного обеспечения, так и на ранних стадиях разработки всего ПТК, но в любом случае одним из первых шагов верификации является определение функции безопасности, подлежащей проверке на корректность.

Функция безопасности представляет собой формализованное условие по отношению к верифицируемой системе, выполнение которого позволяет сделать заключение о безопасности функционирования. Для одного и того же ПТК функция безопасности может быть определена по-разному, а выбор доказываемого условия может происходить на разных этапах жизненного цикла системы. Так, например, функция безопасности может быть определена исходя из функциональности системы; - определение функции безопасности на основании стратегии обеспечения безопасности ко всей системе, требований безопасности к рассматриваемому ПТК и интерфейсов взаимодействия.

Определение и выбор функций безопасности имеет ряд особенностей и проблем. Прежде всего независимое определение функций безопасности используется для последующего анализа корректности предоставленных разработчиками доказательных документов. Исходный код ПО и системные решения ПТК являются опорной информацией, а их анализ может сформировать поведенческую функцию, противоречащую спецификациям. Во время анализа ПО на безопасность может оказаться, что принятая функция безопасности не является необходимой или достаточной, когда заданная функция безопасности слишком строга и доказательство безопасности невозможно, или напротив, слишком слаба, из-за чего уменьшается вероятность нахождения ошибок ПО.

Основными способами изменения функции безопасности является ее расширение (ослабление, более слабое определение) и сужение (усиление, более строгое определение). Кроме этого, функция безопасности может быть изменена не как строгое ослабление или усиление, но в любом случае, она должна находиться в рамках допускаемого безопасного поведения. Однако в случае анализа на безопасность, когда невозможно доказать корректность в предложенном виде или отсутствуют ресурсы для проведения такого объема работ, возможно ослабление проверяемой функции при условии, что это позволит сделать заключение о безопасности ПО.

Разработка и эксплуатация ПО, говорят о том, что чем позже обнаруживается ошибка, тем сложнее как выявить ее, так и исправить, и тем больше проблем она может принести. При этом исправление ошибок, допущенных при формулировании требований к системе, обходится в десятки раз дороже ошибок, допущенных во время реализации [17]. Определение функции безопасности, которое относится к формализации решаемой задачи, является спецификацией по отношению к доказательству корректности и обладает теми же свойствами, что и постановка требований при разработке ПО. Потенциальные ошибки, допускаемые при определении данной функции, негативно влияют на качество верификации и могут приводить к искажению результатов доказательства

корректности и, как следствие, его полному пересмотру. На рис. 2 представлена последовательность этапов анализа на безопасность с определением функции безопасности.

Условия для определения функций безопасности устанавливаются на этапе валидации на основании характеристик применяемых компонентов, стратегий обеспечения безопасности и имеющегося опыта. Данный процесс независим от последующей верификации – он определяет подлежащие проверке свойства и формирует исходные данные, на основании которых задается функция безопасности, используемая в доказательстве корректности. Если допускаются ошибки на этапе валидации, либо не принимаются во внимание особенности поведения, влияющие на безопасность, то это непосредственным образом влияет на качество последующей верификации, а соответственно на качество ПО и ПТК. Кроме того, какие бы эффективные и диверситетные методы и средства не использовались во время доказательства корректности, они не в состоянии выявить и решить проблемы, созданные во время проектирования, так как они работают с одинаковой спецификацией, и только конечный пользователь может указать на ошибку, допущенную при составлении требований.

При проектировании систем могут приниматься неявные допущения, которые напрямую не связаны с безопасностью функционирования, но могут повлиять на работу всего ПТК. Условия безопасности работы системы могут отличаться при изменении окружения или условий функционирования. Таким образом, одной из проблем валидации является определение условий, подлежащих проверке, а особенности данного процесса таковы, что после формализации нет однозначного критерия и уверенности в том, что утвержденная доказываемая функция является необходимой и достаточной. Во время последующей разработки или анализа на безопасность может быть выяснено, что заданные рамки слишком строги и доказательство корректности провести невозможно, или напротив, слишком слабы, из-за чего уменьшается вероятность нахождения ошибок в ПО.

3 Формализация функции безопасности

Валидация представляет собой процесс независимого определения функции безопасности, которая в дальнейшем подлежит верификации. Проблемы валидации заключаются в том, что созданные спецификации на рассматриваемую систему не всегда верны, их ошибки часто проявляются, наносят значительный ущерб и их сложно исправить.

Опыт верификации систем важных для безопасности АЭС выявил ряд ситуаций, когда задание функций безопасности в формализованном виде для некоторых свойств не представляется возможным. В этом случае решением является формализованное описание свойств рассматриваемого понятия в качестве задачи доказательства корректности и определения функции безопасности, а в дальнейшем делается экспертное заключение о том, является ли верифицированное свойство безопасным или нет. Для этого сначала посредством доказательства корректности определяются формализованные свойства системы, а в последующем, на их основании, делается вывод о безопасности системы. Используем три уровня формализации:

1. неформализованный,
2. формализованный,
3. проверяемый.

Определение уровня формализации можно в соответствии с алгоритмом, представленным на рис. 1.

Первый уровень является вербальной формулировкой. Его недостатком является то, что необходимый впоследствии переход к формальному уровню неоднозначен, что может вести к проблемам безопасности и сложностям при доказательстве. Поэтому переход ко второму уровню делать необходимо, и выполнять его как можно раньше.

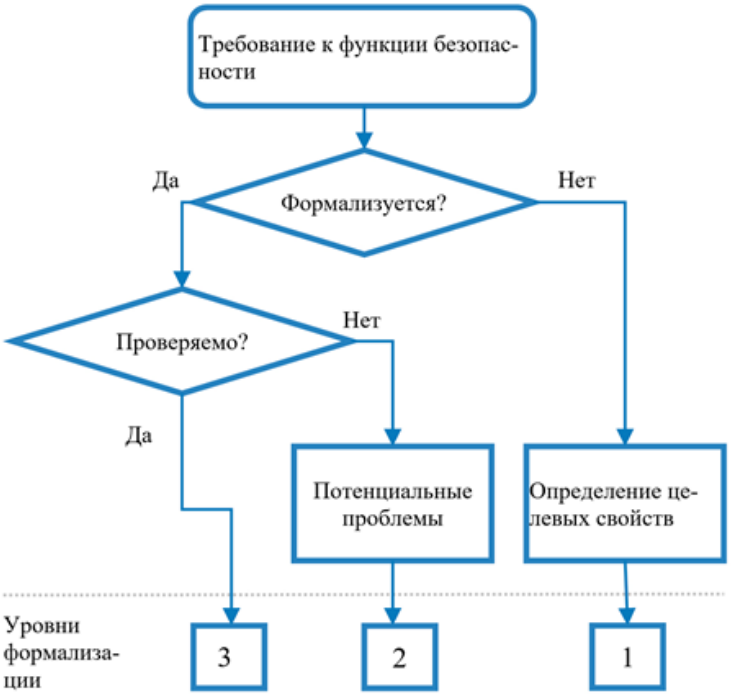

Рис. 1. Определение уровня формализации требования к безопасности

Первый уровень является вербальной формулировкой. Его недостатком является то, что необходимый впоследствии переход к формальному уровню неоднозначен, что может вести к проблемам безопасности и сложностям при доказательстве. Неформализованный уровень появляется изначальным при формулировке, удобен в общении, не требует существенных затрат и является абстрактным. Кроме того, он широко используется в нормативных документах. Тем не менее, в случае рассмотрения конкретной системы и доказательства ее корректности требуется формализация, которая устраняет неоднозначности, улучшает понимание и может быть помещена в некоторую формальную систему для последующего доказательства корректности. Формализованный уровень обладает тем свойством, что он может быть записан в виде знаков некоторой формальной системы. Однако не всегда формализованный вариант может быть проверяемым (полностью или частично), т. е. соответствовать третьему уровню формализации. Данный уровень предполагает, что свойство должно быть фальсифицируемо в рамках имеющихся ресурсов для доказательства. Отсутствие возможности выполнения проверки может быть обусловлено сложностью системы или формулировки, ограниченностью ресурсов доказательства безопасности или другими факторами. Доказательство корректности может работать со вторым уровнем формализации, но отсутствие возможности проверки или ее ограниченность сигнализирует о потенциальных проблемах, так как возможно будет затруднительно использовать другие способы верификации, такие как тестирование, имитационные испытания и др. Сама же необходимость перехода на третий уровень согласуется с опытом создания надежных и безопасных систем.

Любое доказательство всегда базируется на некотором множестве утверждений (аксиом), которые заведомо всегда выполняются. В последующем, на основании аксиом и с помощью правил вывода (логики), проводится доказательство (доказывается теорема) и делается вывод о факте выполнения целевых свойств системы, которые могут выполняться, не выполняться, или теорема может оказаться слишком сложной для доказательства. В качестве аксиом выбираются такие утверждения, которые в наибольшей степени являются неизменными и устойчивыми. Также данные утверждения должны соответствовать рассматриваемому уровню абстракции.

Поэтому переход ко второму уровню делать необходимо, и выполнять его как можно раньше. В общем случае разработка математических идей, к которым относится доказательство корректности, сопровождается аксиоматизацией. Она уточняет понятия, выявляет потенциальные неточности и ошибки, детализирует и формулирует черты нового понятия или абстракции. Строгая формулировка аксиом является одним из мощных средств в борьбе с программными ошибками на всех стадиях жизненного цикла.

4 Результаты верификации функций безопасности

По результатам доказательства корректности необходимо сделать заключение о выполнении ФБ, т. е. обнаружены ли ошибки ПО в процессе валидации и верификации. Это является основным результатом. Однако анализ на безопасность с помощью доказательства корректности обладает рядом особенностей, которые позволяют получить дополнительную выгоду от выполненной работы:

1) результат анализа для обнаруженных ошибок позволяет сформулировать условия, при которых они возникают.
2) одним из результатов является определение спецификаций системы как следствие верификации. Например, может быть определено время реакции на конкретные воздействия, формализовано поведение системы при определенных условиях и др. Это позволяет более точно оценить достоинства и недостатки исследуемого ПТК, его качественные характеристики, рассмотреть детально особенности функционирования.
3) возможность повышения качества ПО в будущем. Рекомендации основываются на анализе, и они могут способствовать улучшению не только надёжности и безопасности системы, но и скорости её работы, устойчивости к сбоям, снижению требований к системе, улучшению качественных характеристик всего исследуемого ПТК.

Стоит отметить, что изложенный подход построен на анализе доказательств функциональной безопасности, а для применения в сфере информационной и кибер-безопасности необходима адаптация подхода. Основным инструментом в подходе является доказательство корректности, которое предоставляет собой доказательство некоторых свойств системы. Соответственно, если они сформулированы в других терминах, таких как надежность или отказоустойчивость ПТК, информационная или кибер- безопасность, способность выдерживать нагрузки и т.д., то для них возможно проведение верификации аналогичным образом. При этом необходимо провести анализ свойств ПО для новых типов задач, т.е. пересмотреть этапы формирования доказываемой функции во время валидации.

Заключение

Определение функции безопасности для проведения верификации является важным этапом анализа на безопасность и ее выбор представляет собой компромисс между имеющимися ресурсами и доказываемыми свойствами. Практика проведения работ по верификации и обеспечения качества программного обеспечения показывает, что избежать компромисса удается только в случае, если система подготовлена к тому, чтобы быть верифицируемой, когда задачи определения функции безопасности (в том числе информационной и кибер- безопасности) решаются до этапов разработки и проектирования, что возможно только для разрабатываемых и проектируемых ПТК.

Для ПТК, используемых в системах важных для безопасности АЭС, существует задача обеспечения правильного (в отношении спецификации), безопасного и полного выполнения требований. Обоснование безопасности системы, безопасности и целостности определенного программного обеспечения основывается на проектировании и проектных документах, представленных во время разработки системы, результате анализа спецификаций, алгоритмов и реализации. Подход, к определению функций безопасности, представленный в статье, был применен:

– при верификации программного обеспечения систем верхнего уровня АСУ ТП АЭС, относящихся к системам важным для безопасности;
– для выявления ошибок проектирования программного обеспечения на ранних стадиях разработки с целью снижения рисков возникновения нештатных ситуаций в процессе эксплуатации объектов;
– при обосновании качества программного обеспечения систем важных для безопасности АЭС на всех этапах жизненного цикла, и позволил повысить качество разрабатываемого/ модифицированного программного обеспечения.

Литература

1. *Poletykin A., Jharko E., Mengazetdinov N., Promyslov V.* Some Issues of Creating the New Generation of Upper Level Control Systems of NPP APCS // Proceedings of the 5th International Conference on Control, Instrumentation, and Automation (ICCIA 2017, Shiraz, Iran). 2017. – P. 78-83.

2. *Менгазетдинов Н. Э., Полетыкин А.Г., Бывайков М.Е., Промыслов В.Г., Жарко Е.Ф., Смирнов В. Б., Акафьев К.В.* Автоматизация атомных электростанций – опыт ИПУ РАН // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014. Москва, 16-19 июня 2014 г. М.: Институт проблем управления им. В. А. Трапезникова РАН, 2014. – С. 4219-4236.
3. *Коган И.Р., Полетыкин А.Г., Промыслов В.Г., Жарко Е.Ф.* Эволюция АСУТП АЭС для ВВЭР, проблемы, нерешенные вопросы, новые угрозы и возможные направления развития // Труды XII Всероссийского совещания по проблемам управления ВСПУ-2014. Москва, 16-19 июня 2014 г. М.: Институт проблем управления им. В. А. Трапезникова РАН, 2014. – С. 4200-4211.
4. *Бывайков М.Е., Жарко Е.Ф., Менгазетдинов Н. Э., Полетыкин А.Г., Прангишвили И.В., Промыслов В.Г.* Опыт проектирования и внедрения системы верхнего блочного уровня АСУ ТП АЭС // Автоматика и телемеханика. 2006. №. 5. – С. 65-79.
5. *Restu Maeran, Joyce Kemunto Mayaka, Jae Cheon Jung*. Software verification process and methodology for the development of FPGA-based engineered safety features system Author links open overlay panel // Nuclear Engineering and Design. Vol. 330. 2018. – P. 325-331.
6. Ye Cheng, Ni Chao, Zheng Tian, Zhang Zhicheng, Zhang Ronghua. Quality assurance for a nuclear power plant simulator by applying standards for safety-critical software // Progress in Nuclear Energy. Vol. 70. 2014. – P. 128-133.
7. *Heung-seop Eoma, Gee-yong Park, Seung-cheol Jang, Han Seong Son, Hyun Gook Kang*. V&V-based remaining fault estimation model for safety–critical software of a nuclear power plant // Annals of Nuclear Energy. Vol. 51. 2013. – P. 38-49.
8. *Drew J. Rankin, Jin Jiang*. A Hardware-in-the-Loop Simulation Platform for the Verification and Validation of Safety Control Systems // IEEE Transactions on Nuclear Science. 2011. Vol. 58, No. 2. – P. 468-478.
9. *Hill J., Tilley S.* Creating safety requirements traceability for assuring and recertifying legacy safety-critical systems // Proceedings of the 18th IEEE International Requirements Engineering Conference. Sydney, Australia, September 27 - October 1, 2010. – P. 297-302,
10. *Leveson N.G., Cha S.S., Shimeall T.J.* Safety verification of ADA programs using software fault trees // IEEE Software. 1991. Vol. 8, No. 4. P. 48-59.
11. *Bozzano M., Villafiorita A., Kerlund O., Bieber P., Bougnol C., Bde E., et. al.* ESACS: An integrated methodology for design and safety analysis of complex systems // Proceedings of the European Safety and Reliability Conference (ESREI 2003). – P. 237-245.
12. *Жарко Е.Ф.* Проблемы управления качеством программного обеспечения // Труды II Международной конференции “Идентификация систем и задачи управления” SICPRO ’03. Москва, 29 -31 января 2003 г. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2003. – С. 887-923.
13. *Joshi A., Miller S.P., Whalen M., Heimdahl M.P.E.* A proposal for model-based safety analysis // Proceedings of the Digital Avionics Systems Conference, DASC. Vol. 2. 2005.– P. 13,
14. *Akerlund O., Bieber P., Boede E., Bozzano M., Bretschneider M., Castel C., Cavallo A. et al.* ISAAC, a framework for integrated safety analysis of functional, geometrical and human aspects // Proceedings of 3rd European Congress on Embedded Real-Time Software. (ERTS’06), 25-27 January 2006. Toulouse, France, 2006.
15. *Smith D., DeLong T., Johnson B.W.* A Safety Assessment Methodology for Complex Safety-Critical Hardware/Software Systems // International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human-Machine Interface Technologies. Washington, DC, November 2000.
16. *Jharko E.Ph.* Towards Quality Assurance under Developing Safety Important Systems Software for Nuclear Power Plants // Proceedings of 2018 International Russian Automation Conference (RusAutoCon). IEEE, 2018. – P. 1-6.
17. *Jharko E.* Towards the quality evaluation of software of control systems of nuclear power plants: Theoretical grounds, main trends and problems // Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics. Colmar, France, July 21-23, 2015. – P. 471-478,
18. *Jharko E.Ph.* Evaluation of the Quality of a Program Code for High Operation Risk Plants // IFAC Proceedings Volumes. Vol. 47, No. 3. 2014.– P. 8060-8065.

19. *Souri A., Navimipour N.J., Rahmani A.M.* Formal verification approaches and standards in the cloud computing: A comprehensive and systematic review // Computer Standards & Interfaces. Vol. 58. 2018. – P. 1-22.
20. *Linna Pang, Chen-Wei Wang, Mark Lawford, Alan Wassyng*. Formal verification of function blocks ap-plied to IEC 61131-3 // Science of Computer Programming. Vol. 113, Part 2. 2015. – P. 149-190.