

МЕТОД АВТОМАТИЧЕСКОГО ФОРМИРОВАНИЯ ТЕЛЕКОММУНИКАЦИОННЫХ МОДУЛЕЙ СТРУКТУРНЫХ ЭЛЕМЕНТОВ АВТОМАТИЗИРОВАННЫХ СИСТЕМ НА ОСНОВЕ XML-ОПИСАНИЯ

А. А. Молев^{а, 1}, адъюнкт

^аНИИИ РЭБ, ВУНЦ ВВС «Военно-воздушная академия им. профессора Н. Е. Жуковского и Ю. А. Гагарина», Воронеж, РФ

Постановка проблемы: в ходе проектирования, разработки и эксплуатации сложных автоматизированных систем, в том числе военного назначения, нередко возникают задачи, когда элементы сложной системы разделяются на две группы: часть представляется реальными аппаратными средствами, а недостающая часть должна быть воспроизведена имитационно. **Цель:** разработка метода автоматического формирования телекоммуникационных модулей, имитирующих недостающие информационные связи структурных элементов автоматизированных систем, гибко конфигурируемых в зависимости от условий применения и формата обмена данными. **Результаты:** предложен метод имитации информационных связей структурных элементов; описана структура унифицированного программного модуля имитации; выделены общие модули, участвующие в процессе обмена информацией и вводе/отображении данных. Приведены требования к унифицированному описанию протокола информационного обмена в формате XML; разработаны алгоритмы обработки входных и выходных данных, основанные на рекурсивном выполнении операций обработки в соответствии с форматом описания; представлены обобщенные диаграммы приема и обработки данных. Программная реализация выполнена на языке C++ в среде Borland Developer Studio. **Практическая значимость:** разработанный метод может быть использован для имитации недостающих информационных связей элементов автоматизированных систем в ходе проведения испытаний на эффективность по основному назначению или пуско-наладочных работ, а также в случаях, когда необходимо получить предельные или трудно сочетаемые значения отдельных характеристик автоматизированной системы или ее элементов.

Ключевые слова — полунатурное моделирование, автоматизированные системы, имитация структурных элементов, информационные связи, формат XML.

Введение

Появление новых видов вооружения и военной техники, рост значений их тактико-технических характеристик повышает требования к экспериментальной базе по обеспечению возможностей проведения проверок опытных образцов с требуемой точностью, достоверностью и оперативностью. Перспективным направлением развития является разработка комплексов средств автоматизации полигонных испытаний, включающих средства телекоммуникационного обмена данными между источниками измерительной информации, испытываемыми образцами вооружения и военной техники и средствами автоматизации полигонных испытаний [1]. Кроме того, при проектировании, разработке и эксплуатации сложных автоматизированных систем (АС), в том числе военного назначения, нередко возникают задачи полунатурного моделирования, когда элементы сложной системы разделяются на две группы: часть структурных элементов сложной системы представлена реальными аппаратными

средствами, а недостающая часть должна быть воспроизведена имитационно [2]. Частичная замена составных частей сложной АС имитационными элементами может применяться:

— при проведении испытаний системы на эффективность по основному назначению, когда натурный эксперимент не может быть реализован в полном объеме из-за отсутствия отдельных элементов;

— для отладки программного обеспечения (ПО) опытных образцов АС, при необходимости оценки функционирования при предельных или трудно сочетаемых значениях отдельных характеристик;

— при замене (изъятии) структурных элементов в процессе функционирования АС для ремонта, доукомплектования или модернизации;

— при модернизации функций системы, требующей уточнения содержания и структуры протоколов обмена информацией.

В этой ситуации существенно упрощаются операции по отладке подсистемы управления, сопряжению составных элементов как на этапе пуско-наладочных работ, так и в ходе испытаний, а также сокращаются затраты на их проведение. Данный подход положен в основу метода полунатурных испытаний [2] и опытно-теоретического метода оценки эффективности сложных систем вооруже-

¹ Научный руководитель — кандидат технических наук, доцент, начальник отдела — заместитель начальника управления НИИИ РЭБ Военно-воздушной академии им. профессора Н. Е. Жуковского и Ю. А. Гагарина И. В. Зайцев.

ния [3], теоретические положения которого описаны в работах [4, 5]. Ключевым фактором, определяющим их применимость, является знание структуры исследуемой автоматизированной системы [6], т. е. состава элементов (подсистем), их функций, внутренних и внешних связей, их характера, а также степени влияния операций, выполняемых элементами и подсистемами, на функционирование системы в целом. Преимуществом данных методов является возможность оценки как отдельных характеристик элементов испытываемой системы, так и характеристик системы в целом с одновременным решением задачи рационального планирования натуральных экспериментов [1].

Во многих случаях ПО имитации структурных элементов АС реализуется, как правило, разработчиками тех же структурных элементов. Также возможны ситуации, когда разработка основного ПО и имитаторов может осуществляться независимо друг от друга с использованием имеющихся протоколов, определяющих структуру, правила и способы обмена информацией в автоматизированной системе [3]. Все это повышает материальные затраты при разработке, а также снижает эффективность работы различных коллективов разработчиков, поскольку требует дополнительного этапа стыковки и отладки ПО имитаторов.

Целью данной работы является разработка метода автоматического формирования телекоммуникационных модулей, имитирующих недостающие информационные связи структурных элементов АС, гибко конфигурируемых в зависимости от условий применения и формата обмена данными. В результате его применения повышается уровень автоматизации процесса испытаний или пуско-наладочных работ АС, а также упрощается оценка значений характеристик АС.

Имитация структурных элементов АС

Взаимосвязи структурных элементов в АС формируются путем сопряжения взаимодействующих объектов друг с другом. Технической основой этого процесса является использование единых требований по обеспечению организационной, технической, программной и информационно-лингвистической совместимости при организации взаимодействия между сопрягаемыми объектами с применением протоколов, определяющих структуру, правила и способы обмена информацией в АС.

При отсутствии отдельных структурных элементов их информационные потоки могут быть воспроизведены с использованием имитатора, реализованного на ЭВМ и имитирующего структуру и процесс функционирования недостающего элемента. Наличие такого имитатора повышает

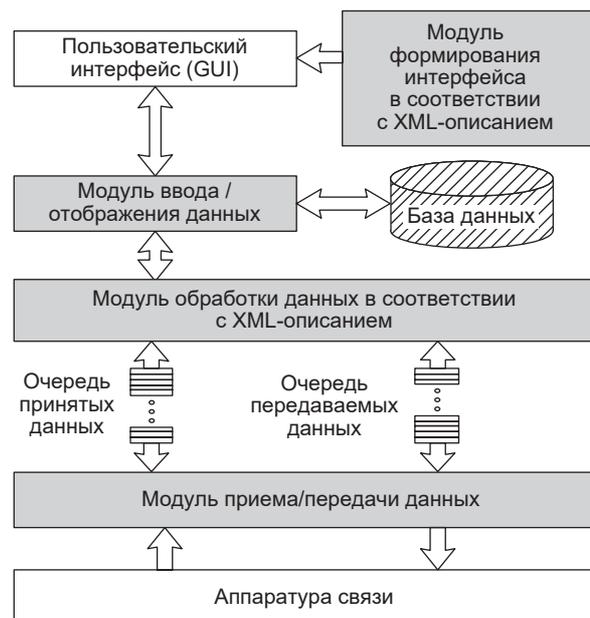
гибкость в выборе условий проведения испытаний, упрощает ограничения в наборе статистического материала. При моделировании появляется возможность многократно повторять реализацию при близких значениях входных данных, что трудно осуществить при проведении сложных натуральных экспериментов.

Для эффективного функционирования имитатора важным фактором является воспроизведение недостающих информационных связей, как правило, формируемых вручную для каждого вновь разрабатываемого ПО имитатора, что при наличии объемного и высокоструктурированного протокола информационно-технического сопряжения является трудоемкой задачей и требует выполнения отдельных операций по отладке и стыковке взаимодействующих элементов.

Для повышения эффективности разработки ПО имитаторов может применяться метод, позволяющий автоматически формировать модули имитации структурных элементов АС.

Необходимо учитывать, что особенности программной реализации имитатора зависят от функций, выполняемых АС, и для систем различного функционального назначения существенно отличаются друг от друга, при этом независимо от назначения можно выделить в них общие модули, участвующие в процессе обмена информацией и вводе/отображении данных (рис. 1).

Сущность предлагаемого метода заключается в автоматическом формировании телекоммуникационных модулей, воспроизводящих функционирование недостающих структурных элемен-



■ **Рис. 1.** Телекоммуникационный модуль имитации недостающих информационных связей структурного элемента АС

тов АС. Для их автоматического формирования используется унифицированный программный модуль и XML-описание, соответствующее заданному формату обмена данными.

В соответствии с приведенной структурой программы при ее функционировании должно обеспечиваться:

- формирование интерфейса ввода и отображения данных;
- подготовка исходных данных для отправки путем заполнения соответствующих полей базы данных как в ручном, так и в автоматизированном режимах;
- передача сообщений информационного обмена вручную и (или) автоматически в соответствии с алгоритмом функционирования;
- автоматический прием и отображение поступающей информации;
- настройка параметров канала связи и характеристик сетевого (транспортного) уровня.

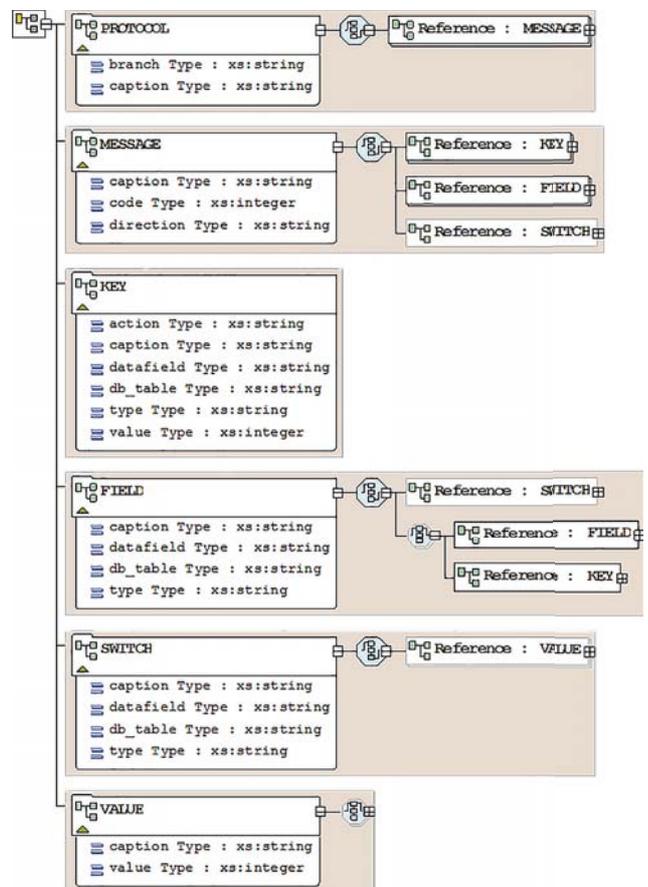
Программная реализация предлагаемого метода формирования модулей имитации структурных элементов АС выполнена в интегрированной среде разработки Borland Developer Studio [7]. Дальнейшим развитием данного метода является создание программного компонента, позволяющего автоматически формировать интерфейс коммуникационного модуля в соответствии с имеющимся описанием.

Данный подход наиболее эффективно реализуется при использовании среды разработки Eclipse на языке Java, в которой возможно в ходе исполнения программы гибко формировать [8] сложно структурированный интерфейс с различными элементами управления: полями ввода данных, радиокнопками, кнопками-флажками, таблицами и др. Однако в данном случае требуется изменить исходный код с языка C++ на язык Java, что, с одной стороны, является трудоемкой операцией, с другой стороны, расширяет гибкость использования разработанной программы и позволяет воспользоваться кроссплатформенностью языка Java и не ограничиваться рамками операционной системы Windows, а иметь возможность запускать разработанную программу в любой современной операционной системе, поддерживающей работу в виртуальной машине Java.

Основой для формирования модулей является использование унифицированного описания протоколов обмена данными в формате XML (eXtensible Markup Language). С помощью XML-описания автоматически генерируется интерфейс ввода/отображения принимаемой и передаваемой информации, а также чтение и запись в базу данных. Использование XML-формата обуславливается его удобством для описания информации, имеющей заранее определенную структуру. Унифицированное описание данных

в формате XML также используется в технологии сериализации данных [9], которая в настоящем случае не может быть применима, поскольку предусматривает хранение данных в полях классов, описывающих форматы, типы и структуру данных. Однако данные классы разрабатываются на этапе написания программного кода и не позволяют гибко изменять форматы и типы данных во время эксплуатации имитационных модулей, требуя перекомпиляции программы. В качестве альтернативы предлагается метод, основанный на описании структуры и формата передаваемых данных без учета особенностей программной реализации конкретных объектов.

Для задания форматов и содержания сообщений протокола информационного обмена используется XML-описание, структура которого в виде диаграммы XSD представлена на рис. 2. На верхнем уровне располагается элемент *protocol*, включающий в себя совокупность элементов *message*, каждый из которых соответствует отдельному информационному сообщению. В состав каждого сообщения входят элементы двух видов — *key* и *field*, соответствующие ключевым и информационным полям таблиц базы данных. Элементы



■ Рис. 2. XSD-диаграмма обобщенного описания протокола информационного обмена

Элемент	Атрибуты и их значение
<i>protocol</i>	<i>branch</i> — наименование автоматизированной системы <i>caption</i> — наименование протокола обмена
<i>message</i>	<i>caption</i> — наименование сообщения <i>code</i> — код/идентификатор сообщения <i>direction</i> — вид сообщения (входящее, исходящее, двунаправленное)
<i>key/field</i>	<i>caption</i> — наименование поля <i>db_table</i> — наименование таблицы базы данных <i>datafield</i> — наименование поля в таблице базы данных <i>type</i> — тип поля (целое, вещественное, дата и время, строка, составной тип, счетчик) <i>value</i> — значение, указанное в XML-описании (используется в сочетании с атрибутом <i>action</i> , равным <i>default_value</i>)
Только для элемента <i>key</i>	<i>action</i> — действие при чтении/записи значения поля: <i>increment</i> — увеличение счетчика <i>read_from_data</i> — чтение из принятых данных <i>get_DT</i> — запись даты, времени создания сообщения <i>default_value</i> — запись значения, заданного в XML-описании
<i>switch</i>	Селектор выбора элемента XML-описания в зависимости от значения дочернего элемента <i>value</i> , аналог поля <i>field</i> , содержащего целое значение <i>caption</i> — наименование поля <i>db_table</i> — наименование таблицы базы данных <i>datafield</i> — наименование поля в таблице базы данных <i>type</i> — тип поля (для элемента <i>switch</i> используются только целочисленные поля — 1, 2, 4 байта и т. д.)
<i>value</i>	<i>caption</i> — наименование поля <i>value</i> — значение, соответствующее сохраненному в поле <i>datafield</i> таблицы <i>db_table</i> элемента <i>switch</i> и определяющее использование последующих элементов XML-описания

field имеют структуру, определяемую форматом передаваемых данных, и в случае вложенных сообщений включают те же элементы — *key* и *field*. Информационный обмен ведется сообщениями, законченными в смысловом отношении и имеющими произвольную длину. Структура сообщений определяется соответствующими протоколами информационного сопряжения.

Значение атрибутов каждого элемента приведено в таблице.

При формировании XML необходимо учитывать следующее:

- структура описания должна строго соответствовать формату протокола обмена;

- наименования таблиц и полей должны совпадать с соответствующими таблицами и полями базы данных;

- первоначально приводится информация о ключевых полях таблицы базы данных, далее — поля данных;

- при записи принимаемой информации в базу данных сначала пишутся ключевые поля, далее для той же записи сохраняются все оставшиеся поля значений;

- при наличии вложенных данных запись значений ключевых полей осуществляется поэтапно в соответствии с иерархией в порядке: ключевые поля и поля данных (при наличии)

главной таблицы, ключевые поля и поля данных (при наличии) подчиненной таблицы и т. д.;

— значения ключевых полей таблиц могут формироваться как автоматически, так и считываться из принимаемых данных.

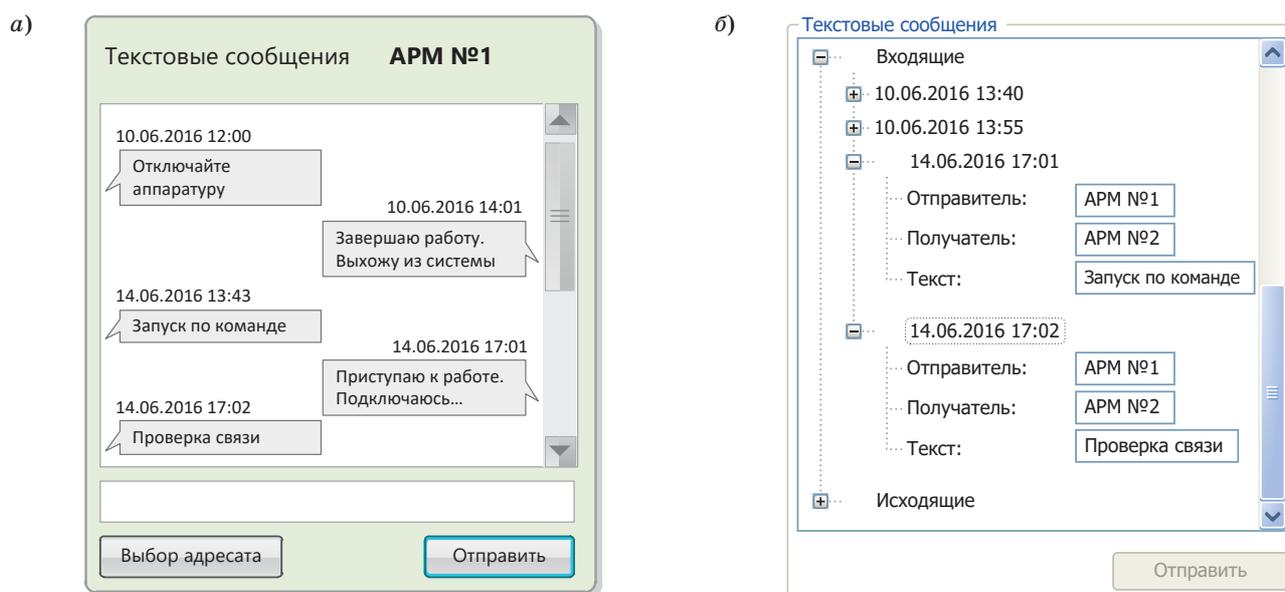
В качестве примера приведено XML-описание текстового сообщения, передаваемого по каналу связи в формате:

Отправитель	Получатель	Текст

имеющее вид

```
<message caption="Текстовое сообщение" code="30">
  <key action="get_DT" caption="" datafield="DT" db_
table="messages" type="DT"/>
  <key caption="Вид сообщения" datafield="type" type="int"
action="default_value" db_table="messages" value="0"/>
  <field caption="Отправитель" datafield="from" db_
table="messages" type="text"/>
  <field caption="Текст" datafield="text" db_table="messages"
type="text"/>
</message>
```

Помимо описания непосредственно информационных полей, здесь также применяются дополнительные элементы *key* для задания ключевых полей «дата, время» и «вид сообщения».



■ **Рис. 3.** Пример интерфейса ввода/просмотра текстовых сообщений: *а* — спроектированного на этапе разработки программы; *б* — автоматически сформированного в процессе функционирования

Для обозначения элементов XML-описания протокола далее будут применяться следующие наименования:

Message — элемент (узел дерева XML), соответствующий отдельному сообщению;

Node — текущий элемент;

ChildField — дочерний элемент, подчиненный текущему.

Ввод и отображение данных могут быть реализованы с использованием как заранее подготовленного интерфейса, так и динамически формируемого при открытии соответствующего окна. Первый вариант предпочтителен, если разрабатываются имитаторы для имеющегося формата данных либо возможные изменения будут минимальными. Интерфейс в данном случае является более эргономичным и проработанным в деталях, что повышает удобство применения и улучшает внешний вид, а также упрощает работу, поскольку ряд информационных полей может быть скрыт от ввода, и значения для них будут формироваться автоматически. Второй вариант может реализовываться на этапе отладки взаимодействия элементов АС, проведении стыковочных работ. Здесь применяются наиболее общие элементы управления и простая структура построения окон, вопросы удобства и эргономичности учитываются в минимальном объеме. Возможные варианты реализации интерфейса с применением указанных подходов показаны на рис. 3. Видно, что необходимый функционал обеспечивается в обоих случаях, однако во втором варианте интерфейс перегружен однотипными элементами, добавляющими избыточность, а также требуется ввод всех

полей информационного сообщения, в том числе тех, которые могут заполняться автоматически.

Формирование выходных данных

Конкретное содержание и принцип формирования значений информационных полей зависит от задачи, для решения которой используется имитатор. Возможные варианты значений полей могут быть:

- введены оператором вручную;
- загружены из базы данных в соответствии с заранее заданным контрольным примером;
- сгенерированы случайно, в пределах допустимых значений.

Для первых двух вариантов значения полей должны обеспечивать конкретное смысловое содержание передаваемых сообщений для программной совместимости при взаимодействии между объектами в штатном режиме. Третий вариант наполнения может быть использован при передаче потока требуемой информации с высокой интенсивностью для тестирования производительности канала связи или обработчика входных данных на приемной стороне.

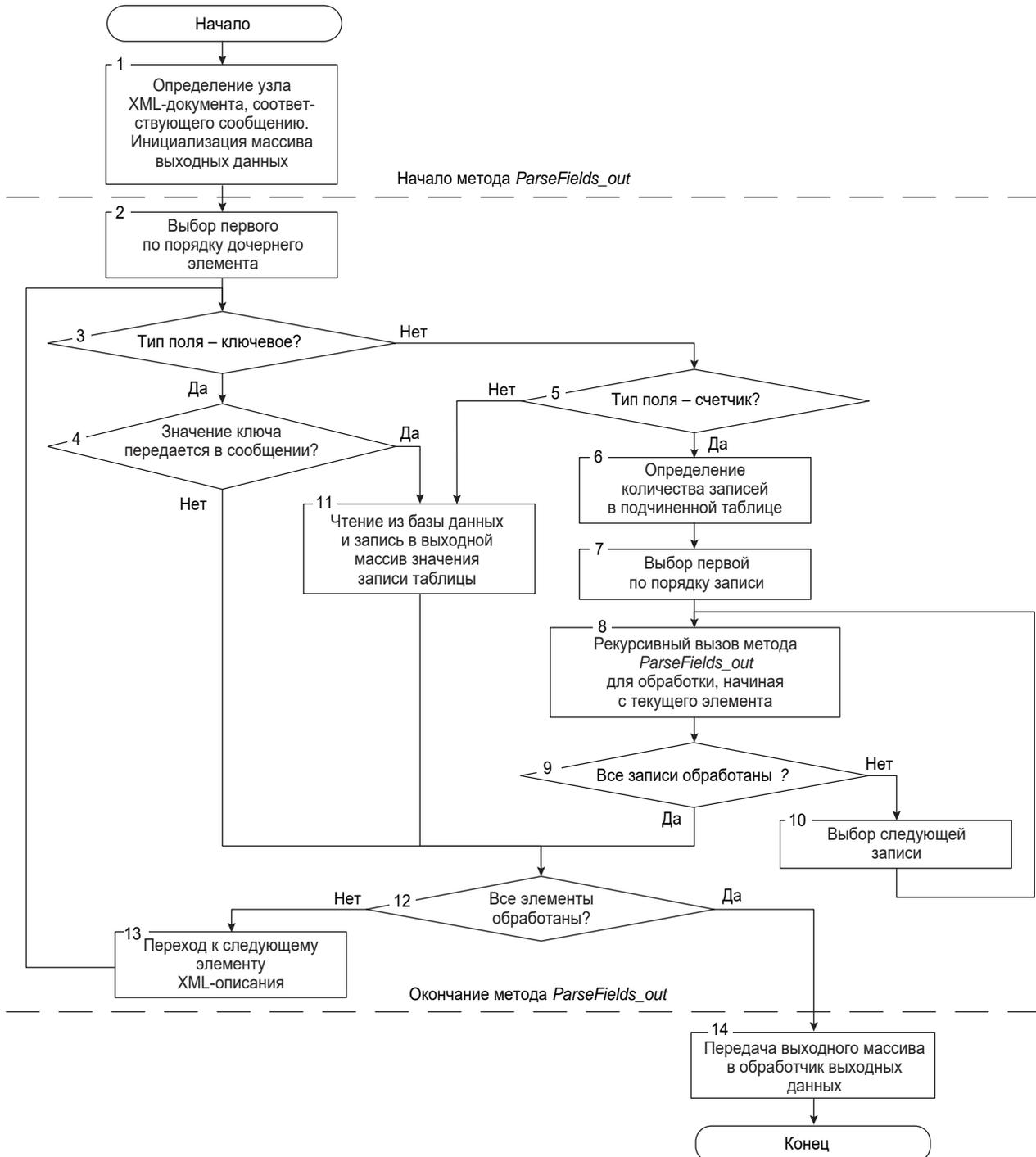
Информационные сообщения формируются путем преобразования записей базы данных в соответствии с форматом протокола. На основе XML-описания путем последовательного чтения записей из таблиц базы данных формируется выходной массив байтов, далее передаваемый в модуль обработки выходных данных для «упаковки» в формат транспортного и сетевого протоколов и последующей передачи по каналу связи.

Поскольку в самом общем случае сообщения имеют структуру произвольной степени вложенности, для обработки XML-описания используются рекурсивные вызовы соответствующих методов, позволяющие автоматически сформировать (обрабатывать при приеме) линейный набор данных в заданном формате.

Блок-схема алгоритма формирования выходного массива приведена на рис. 4. Фрагмент ал-

горитма, начиная с п. 2 и до п. 13 включительно, относится к методу *ParseFields_out*, формирующему массив выходных данных *byte* data*, описание которых соответствует заданному узлу *Node*. Алгоритм формирования массива передаваемых данных заключается в выполнении следующих операций.

1. Определяется узел XML-документа *MessageNode*, соответствующий отправляемому со-



■ Рис. 4. Алгоритм формирования массива выходных данных перед отправкой

общению. Инициализация массива выходных данных.

2. Выбор первого по порядку дочернего элемента *ChildField*.

3. Проверка вида элемента *ChildField* — ключевое поле (*key*) или поле данных (*field*). Если поле данных, переход к п. 6.

4. Проверка действия для ключевого поля — необходимо передавать в сообщении (для действия «чтение из данных» *read_from_data*) или нет (действия *increment*, *get_DT*, *default_value*). Если значение ключа не передается в сообщении, то переход к п. 11.

5. Проверка принадлежности поля данных к типу «счетчик». Если условие не выполняется, то переход к п. 11.

6. Определение количества записей *child_count* в подчиненной таблице.

7. Выбор первой по порядку записи.

8. Вызов метода *ParseFields_out* для обработки записей, начиная с текущего элемента *Node*.

9. Проверка завершения обработки всех записей подчиненной таблицы. Если условие выполняется, то переход к п. 12.

10. Выбор следующей записи в таблице, переход к п. 8.

11. Чтение значения поля *datafield* из соответствующей таблицы *db_table*, запись значения в массив выходных данных *data*.

12. Проверка условия обработки всех элементов XML-описания. Если условие выполняется, то переход к п. 14.

13. Выбор следующего элемента, переход к п. 2.

14. Передача выходного массива *data* в обработчик исходящих данных для последующей отправки, завершение работы алгоритма.

Сформированный массив *data* помещается в очередь на передачу данных, откуда извлекается непосредственно перед отправкой. Формирование пакетов для передачи по каналу связи в случае сети Ethernet осуществляется автоматически с использованием стандартных компонентов типа *TCPServer*, *TCPClient* и т. п.

В зависимости от специфики задач, решаемых имитатором, и выполняемых им функций, информационные сообщения могут передаваться:

— оператором вручную;

— автоматически, в ответ на сообщения, требующие реакции, либо немедленно, либо через заданный интервал времени;

— по заранее заданному расписанию либо в соответствии с алгоритмом функционирования; — в случайном порядке для обеспечения загрузки канала связи.

Для заполнения информационных полей сообщений может использоваться информация, введенная заранее в базу данных либо сгенерированная случайным образом перед отправкой. При

имитации элементов системы, когда не требуется или невозможно обеспечить присутствие оператора, предпочтительно использовать автоматические варианты подготовки и отправки сообщений. При наличии оператора ручной или автоматический вариант выбирается по ситуации в соответствии с алгоритмами функционирования.

Обработка входных данных

При установленном соединении между объектами сообщения принимаются автоматически и поступают на обработку в очередь входящих данных. Принятая информация в соответствии с XML-описанием подвергается разбору по процедуре, схожей с выполняемой при отправке данных. Основным отличием является запись в базу данных в первую очередь ключевых полей для сохранения целостности записей, в последующем — запись значений недостающих полей записи, в отличие от поочередной обработки каждого поля, реализуемой при отправке. Блок-схема алгоритма обработки массива входных данных приведена на рис. 5. Фрагмент алгоритма, начиная с п. 2 и до п. 13 включительно, относится к методу *ParseFields_in*, формирующему массив принятых данных, описание которых соответствует заданному узлу *Node*.

Алгоритм обработки входных данных заключается в выполнении следующих операций.

1. Определение узла XML-документа *Message Node*, соответствующего принятому сообщению.

2. Выбор первого по порядку дочернего элемента *ChildField*.

3. Инициализация массива значений ключевых полей переменной длины *Variant* keys*.

4. Проверка вида элемента *ChildField* — ключевое поле (*key*) или поле данных (*field*). Если условие не выполняется, то переход к п. 11.

5. Проверка действия для ключевого поля — передается в сообщении (для действия «чтение из данных» *read_from_data*) или нет (действия *increment*, *get_DT*, *default_value*). Если значение ключа не передается в сообщении, то переход к п. 7.

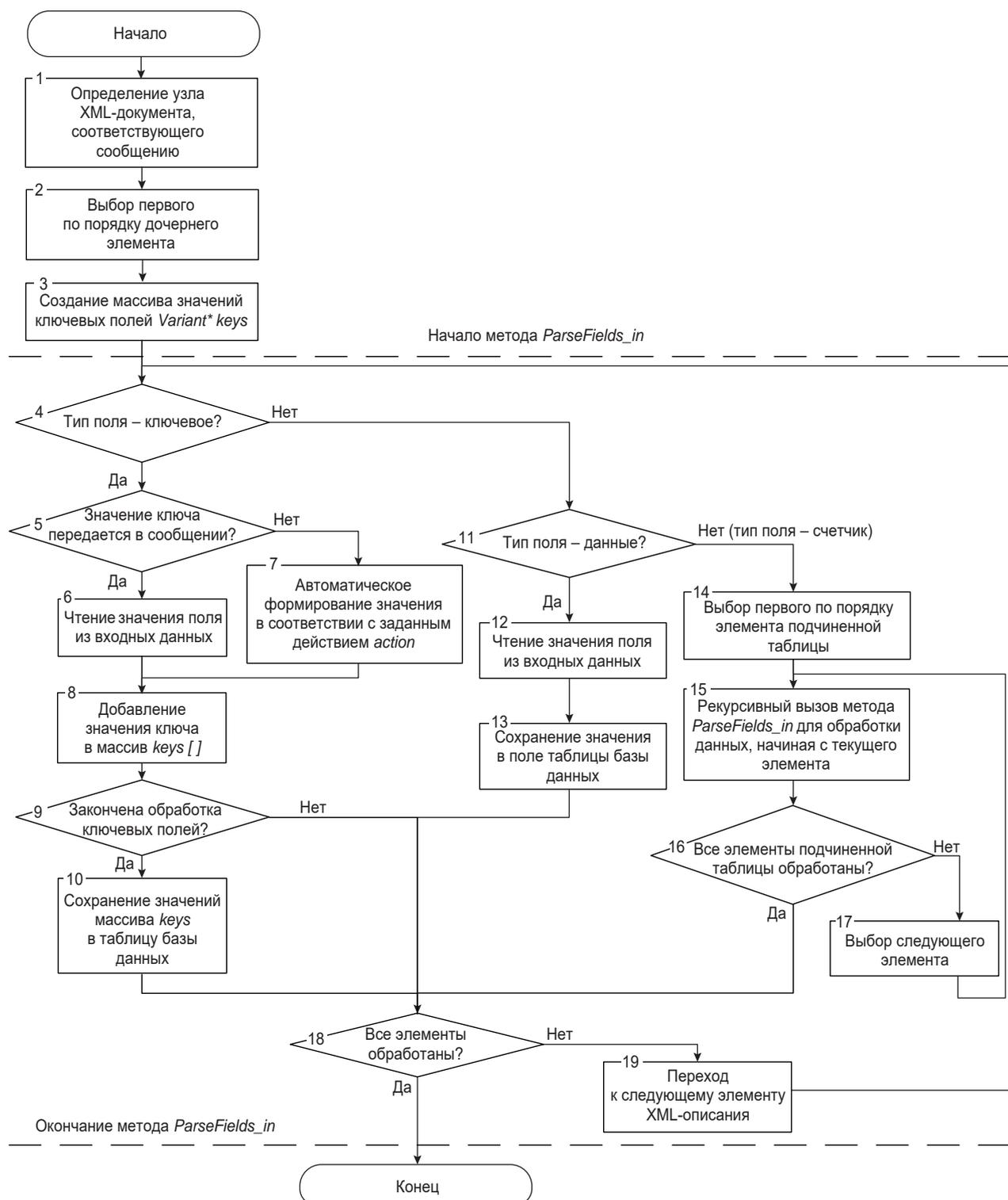
6. Чтение значения ключевого поля из принятых данных в соответствии с его типом *type*.

7. Автоматическое формирование значения ключевого поля в соответствии с заданным действием *action* (см. таблицу значений атрибутов XML-описания).

8. Добавление значения ключевого поля в массив *keys*.

9. Проверка условия окончания обработки всех ключевых полей. Если условие не выполняется, то переход к п. 18.

10. Добавление новой записи в таблицу *db_table*, сохранение значений массива *keys[]*.



■ Рис. 5. Алгоритм обработки входных данных

11. Проверка принадлежности поля *datafield* к типу «данные». Если условие не выполняется, то переход к п. 14.

12. Чтение значения из входных данных в соответствии с его типом.

13. Сохранение значения в поле *datafield* таблицы базы данных *db_table*, затем переход к п. 18.

14. Выбор первого по порядку элемента подчиненной таблицы.

15. Рекурсивный вызов метода *ParseFields_out* для обработки записей, начиная с текущего элемента *Node*.

16. Проверка завершения обработки всех записей подчиненной таблицы. Если условие выполняется, то переход к п. 18.

17. Выбор следующей записи в таблице, переход к п. 8.

18. Проверка условия обработки всех элементов XML-описания. Если условие выполняется, завершение работы алгоритма.

19. Выбор следующего элемента XML-описания, переход к п. 4.

Обмен сообщениями и управление параметрами канала связи

Перечень настраиваемых параметров канала связи и характеристики сетевого (транспортного) уровня передачи определяются типом протокола сопряжения передачи данных сетевого (транспортного) уровня и стыком — RS 232, Ethernet и т. п. При передаче данных по Ethernet, как правило, настраиваются IP-адреса получателей и отправителей сообщений, используемые порты для TCP-сокетов, виды сокетов — сервер или клиент. На каждой из взаимодействующих сторон создаются соответствующие виды сокетов, клиенты подключаются к прослушивающему сокету на заданном порту, т. е. устанавливают соединение и осуществляют двунаправленный обмен информацией. В схематичном виде последовательности выполнения операций частными модулями, участвующими в процессе обмена информацией и вводе/отображении данных, представлены на рис. 6.

При отправке данных (рис. 6, а) по идентификатору сообщения определяется необходимый узел XML-описания, далее, в соответствии с алгоритмом (см. рис. 5), осуществляется формирование массива данных *data*. Сформированный массив данных поступает в очередь на отправку, из которой по ме-

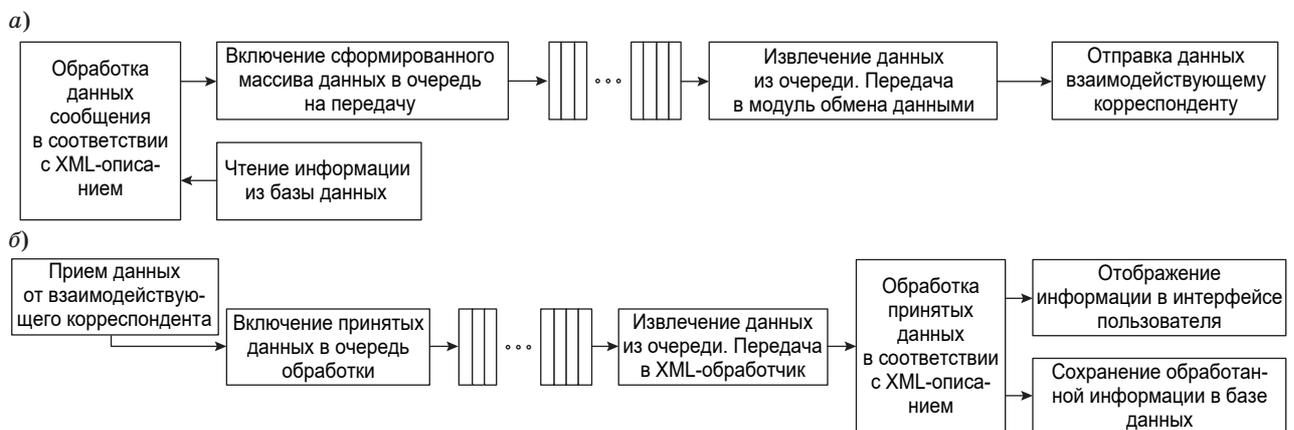
ре ее обработки извлекаются данные. После этого осуществляется их отправка взаимодействующему корреспонденту по организованному и установленному каналу связи. Операции, выполняемые при приеме данных (рис. 6, б), соответствуют аналогичным операциям, выполняемым при отправке, и осуществляются в обратной последовательности.

Заключение

В статье рассмотрены вопросы имитации информационных связей структурных элементов АС на основе их замены унифицированными телекоммуникационными модулями. Приведены структура модуля имитации, требования к формату унифицированного описания структуры информационного обмена, алгоритмы обработки входных и выходных данных, обобщенные алгоритмы работы программных модулей имитации структурных элементов. Унификация реализована путем использования XML-описания, имеющего единый формат для любого структурного элемента независимо от алгоритма его функционирования.

Программная реализация предлагаемого метода выполнена в интегрированной среде разработки Borland Developer Studio. Предложенный метод целесообразно использовать при решении задач полунатурного моделирования, в ходе проведения испытаний либо в случаях, когда необходимо получить предельные или трудно сочетаемые значения характеристик автоматизированной системы или ее элементов.

Таким образом, разработан метод, позволяющий повысить степень автоматизации процесса испытаний автоматизированных систем на эффективность по основному назначению или пуско-наладочных работ за счет воспроизведения функционирования структурных элементов с применением имитаторов, гибко конфигурируемых в зависимости от условий применения и формата обмена данными.



Литература

1. Буренок В. М., Найденов В. Г. Требуется модернизация полигонов // Воздушно-космическая оборона. 2009. № 4. С. 32–39.
2. Молев А. А., Зайцев И. В. Применение агрегативного подхода к моделированию элементов радиоэлектронных систем при проведении опытно-теоретических исследований их эффективности // Имитационное моделирование. Теория и практика: сб. докл. Второй Всерос. науч.-практ. конф. ИММОД-2005. Т. 1. СПб.: ЦНИИТС, 2005. С. 143–147.
3. Шестихин В. И., Салтанов П. Я., Якубовский С. В. Опытнo-теоретический метод испытаний системы предупреждения о ракетном нападении // Военная мысль. 2015. № 6. С. 37–41.
4. Шаракшанэ А. С., Железнов И. Г., Ивницкий В. А. Сложные системы. — М.: Высш. шк., 1977. — 247 с.
5. Бусленко В. Н. Автоматизация имитационного моделирования сложных систем. — М.: Наука, 1977. — 240 с.
6. Савин Г. И. Системное моделирование сложных процессов. — М.: ФАЗИС; ВЦ РАН, 2000. — 276 с.
7. Свидетельство о государственной регистрации программы для ЭВМ № 2016617747 от 14.07.2016. Унифицированный модуль имитации структурных элементов автоматизированных систем / Молев А. А., заявл. 24.05.2016 г.; опубл. 20.08.2016 г., Бюл. № 8. — 1 с.
8. Хемрадждани А. Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse: пер. с англ. — М.: Вильямс, 2008. — 352 с.
9. Хорстманн К., Корнелл Г. Java 2. Библиотека профессионала. 9-е изд. Т. 1. — М.: Вильямс, 2015. — 864 с.

UDC 681.518

doi:10.15217/issn1684-8853.2017.1.40

XML-based Method for Automatic Formation of Telecommunication Modules of Structural Elements in Automated System

Molev A. A.^a, Post-Graduate Student, aadm@mail.ru^aMilitary Education-Science Center of Military Air Forces «Professor N. E. Zhukovsky and Yu. A. Gagarin Military Air Academy», 54A, Staryh Bolshevnikov St., 394064, Voronezh, Russian Federation

Introduction: During the design, development and operation of complex automated systems, including military ones, certain problems often arise when the elements of a complex system are split into two groups. One part is represented by real hardware, and the missing part should be reproduced by simulation. **Purpose:** We develop a method to automatically generate telecommunication modules which simulate the missing informational links between the structural elements of automated systems, flexibly configurable depending on the application and the data exchange format. **Results:** A method is proposed for the simulation of informational links between structural elements. The structure of a unified simulation software units is discussed. The common modules are specified which take part in the communication and data input/output. The requirements are formulated for the XML unified description of the information exchange. The algorithms for processing the input and output data are developed, based on the recursive implementation of the processing operations in accordance with the description format. General charts are provided for data reception and processing. The software was made in Borland Developer Studio IDE using C++ language. **Practical relevance:** The developed method can be used to simulate missing informational links of automated systems in the course of testing the main-purpose efficiency or commissioning work, as well as in cases when it is necessary to obtain extreme or difficult to combine values of individual characteristics of an automated system or its components.

Keywords — Hardware-in-the-Loop (HIL) Simulation, Automated Systems, Simulation of Structural Elements, Informational Links, XML Format.

References

1. Burenok V. M., Naidenov V. G. Modernization of Polygons is Required. *Vozdushno-kosmicheskaja oborona*, 2009, no. 6, pp. 32–39 (In Russian).
2. Molev A. A., Zaitsev I. V. Application of Aggregative Approach to Modeling Elements of Electronic Systems During the Development of Theoretical Studies of Their Effectiveness. *Sbornik dokladov Vtoroi Vserossiiskoi nauchno-prakticheskoi konferentsii IMMOD-2005* [Proc. 2nd Rus. Conf. "Simulation Modeling. Theory and Practice IMMOD-2005"]. Saint-Petersburg, 2005, vol. 1, pp. 143–147 (In Russian).
3. Shestikhin V. I., Saltanov P. Ia., Iakubovskii S. V. Experimental and Theoretical Warning System Test Method of Missile Attack. *Voennaia mysl'*, 2015, no. 6, pp. 37–41 (In Russian).
4. Sharakshane A. S., Zheleznov I. G., Ivnickii V. A. *Slozhnye sistemy* [Complex Systems]. Moscow, Vysshaia shkola Publ., 1977. 247 p. (In Russian).
5. Buslenko V. N. *Avtomatizatsiia imitatsionnogo modelirovaniia slozhnykh sistem* [Automation Simulation of Complex Systems]. Moscow, Nauka Publ., 1977. 240 p. (In Russian).
6. Savin G. I. *Sistemnoe modelirovanie slozhnykh protsessov* [System Modeling of Complex Processes]. Moscow, FAZIS; VTs RAN Publ., 2000. 276 p. (In Russian).
7. Molev A. A. *Unifitsirovannyi modul' imitatsii strukturnykh elementov avtomatizirovannykh sistem* [Unified Simulation Module of the Structural Elements of the Automated Systems]. Program RF, no. 2016617747, 14.07.2016.
8. Hemrajani A. *Agile Java Development with Spring, Hibernate and Eclipse*. Sams Publishing, 2006. 360 p.
9. Horstmann C. S., Cornell G. *Core Java*. Vol. I. *Fundamentals*. 9th Ed. Prentice Hall, 2012. 1008 p.