до продуктивності (SCADA-системи), наявними вимогами кросплатформенності та підвищенними вимогами до безпеки передачі даних (банківські системи, OLAP, ERM/CRM) кращим вибором стане серіалізація за допомогою протокола Protobuf, в якому (на відміну від інших) також присутні високорівневі засоби та інструменти для роботи. Окрім того, кінцевий розмір серіалізованих даних буде мінімальний, а також гарантується цілісність та захищеність даних.

Література

1. PROTOBUF VS. BOOST: SERIALIZATION [Электронный ресурс] // Журнал «Хакер». — 31.10.2013. — Режим доступа: \www/URL: https://xakep.ru/2013/10/31/protobuf-vs-boost-serialization/
2. Еллоіт, Р. XML [Текст] / Р. Еллоіт. — Спб.: Символ-Плюс, 2001. — 576 с.
3. Введение в JSON [Электронный ресурс]. — 11.11.2011. — Режим доступа: \www/URL: http://json.org/json-ru.html
4. Json или «Туда и Обратно» [Электронный ресурс] // Хабрахабр. — 01.08.2014. — Режим доступа: \www/URL: https://habrahabr.ru/company/naumen/blog/228279/
5. Protocol Buffers [Electronic resource] // Google Developers. — Available at: \www/URL: https://developers.google.com/protocol-buffers/. — 05.01.2016.
6. Protocol Buffer Basics: C++ [Electronic resource] // Google Developers. — 03.09.2014. — Available at: \www/URL: https://developers.google.com/protocol-buffers/docs/cpptutorial#why-use-protocol-buffers. — 05.01.2016.
7. Google Protocol Buffers in action (C++) [Electronic resource]. — 16.09.2012. — Available at: \www/URL: http://forums.4fips.com/viewtopic.php?f=3&t=807
8. JSON и XML. Что лучше? [Электронный ресурс] // Хабрахабр. —24.08.2007. — Режим доступа: \www/URL: https://habrahabr.ru/post/31225/
9. 5 Reasons to Use Protocol Buffers Instead of JSON For Your Next Service [Electronic resource] // Code Climate. — 05.06.2014. — Available at: \www/URL: http://blog.codeclimate.com/blog/2014/06/05/choose-protocol-buffers/
10. ProtoBuf.js vs JSON [Electronic resource] // GitHub, Inc. — 02.02.2015. — Available at: \www/URL:https://github.com/dcodeIO/protobuf.js/wiki/ProtoBuf.js-vs-JSON
11. COMPARING PROTOBUF, JSON, BSON, XML WITH .NET FOR FILE STREAMS [Electronic resource] // Software Engineering. — 09.01.2014. — Available at: \www/URL: http://damienbod.com/2014/01/09/comparing-protobuf-json-bson-xml-with-net-for-file-streams/
12. Если вы еще используете JSON, то Google protobuf идет к вам! [Электронный ресурс] // Дневник программиста. — 27.11.2012. — Режим доступа: \www/URL: http://knzsoft.blogspot.com/2012/11/protobuf.html
13. Сопоставление JSON и XML [Электронный ресурс] // Microsoft. — Режим доступа: \www/URL: https://msdn.microsoft.com/ru-ru/library/bb924435(v=vs.110).aspx. — 01.02.2016.
14. Как сериализовать и десериализовать данные JSON [Электронный ресурс] // Microsoft. — Режим доступа: \www/URL: https://msdn.microsoft.com/ru-ru/library/bb412179(v=vs.110).aspx. — 01.02.2016.

ВЫБОР ПРОТОКОЛА СЕРИАЛИЗАЦИИ ПРИ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В данной статье рассмотрены современные протоколы сериализации данных XML, JSON, упаковка в бинарный вид, Protobuf и представление данных в виде строк. Проведен анализ данных способов сериализации данных для дальнейшего использования в разработке программного обеспечения. Описаны основные преимущества и недостатки вышеуказанных протоколов сериализации. Сделаны выводы по целесообразности использования каждого из них.

Ключевые слова: протокол, XML, JSON, Protobuf, сериализация, парсинг, упаковка, бинарный.

Грудзинський Юліан Євгенович, *старший викладач, кафедра автоматизації теплоенергетичних процесів, Національний технічний університет України «Київський політехнічний інститут», Україна,* ***e-mail: jug@sonettele.com****.*

Марков Роман Валерійович, *кафедра автоматизації теплоенергетичних процесів, Національний технічний університет України «Київський політехнічний інститут», Україна.*

Грудзинский Юлиан Евгеньевич, *старший преподаватель, кафедра автоматизации теплоэнергетических процессов, Национальный технический университет Украины «Киевский политехнический институт», Украина.*

Марков Роман Валерьевич, *кафедра автоматизации теплоэнергетических процессов, Национальный технический университет Украины «Киевский политехнический институт», Украина.*

Grudzynskyy Yulian, *National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine,* ***e-mail: jug@sonettele.com****.*

Markov Roman, *National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine*

УДК 004.021+004.052.42
DOI: 10.15587/2312-8372.2016.66444

Дубинский А. Г.,
Хорольский О. А.

РАЗРАБОТКА МЕТОДА ПРОВЕРКИ БЛОК-СХЕМ МЕДИЦИНСКИХ АЛГОРИТМОВ

Обеспечение высокого качества медицинской помощи требует точной и эффективной записи медицинских алгоритмов. Предложен метод для выявления частей схем алгоритмов, которые записаны не в соответствии с государственным стандартом, что допускает их неоднозначное толкование и может стать причиной врачебных ошибок. Дан алгоритм выявления таких частей, показан пример его применения для обнаружения ошибок.

Ключевые слова: *алгоритм, стандарт, блок-схема, клиническое руководство, медицинский стандарт, клинический протокол.*

1. Введение

Под руководством Министерства здравоохранения Украины уже несколько десятилетий ведется работа по созданию полной базы медико-технологических документов по стандартизации медицинской помощи.

Новая система представлена тремя типами документов: унифицированные клинические протоколы медицинской

помощи, стандарты медицинской помощи и адаптированные клинические руководства, рекомендованные как лучшая клиническая практика. В настоящее время в реестре медико-технологических документов собрано более 170 таких документов, утвержденных МОЗ Украины. В базе стандартов медицинской помощи отдельно представлены документы, разработанные до 2012 года, в основном экспертным методом, т. е. не в соответствии с принципами доказательной медицины. Развернутый анализ достигнутых результатов дан в работах [1, 2].

Неотъемлемой частью этих документов являются алгоритмы, в частности алгоритмы действий врача при выполнении диагностики и лечении, а также алгоритмы проведения медицинских манипуляций. Для эффективного использования этих документов в реальных условиях, недостаточно представить описание алгоритмов в текстовом виде. Необходимо визуализировать алгоритмы, дать их графическое представление с помощью схем. Это облегчит понимание алгоритмов и позволит врачу быстро определять, какие действия следует выполнять на каждом этапе работы с пациентом. Точная и понятная схема алгоритма должна способствовать уменьшению частоты врачебных ошибок.

Предварительный анализ схем алгоритмов в утвержденных министерством документах медицинских стандартов и протоколов показал, что их создатели не проверяли результаты своей работы на соответствие требованиям стандартов составления схем алгоритмов [3]. Поскольку методика разработки и внедрения медицинских стандартов предусматривает плановый этап обновления медико-технологических документов, следует выявить несоответствия в схемах алгоритмов с целью их устранения в будущем.

2. Анализ литературных данных и постановка задачи

Существует несколько основных стандартов для графической записи алгоритмов. В постсоветских государствах традиционно используют так называемые «блок-схемы», изучаемые как в школьном курсе информатики, так и в вузовской дисциплине «медицинская информатика», которые основаны на ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем». Этот действующий стандарт разработан методом прямого применения международного стандарта ISO 5807-85. Идея схематической записи алгоритмов получило дальнейшее развитие в виде универсального языка моделирования (UML). UML версии 1.4.2 был принят в 2005 г. Международной организацией по стандартизации как стандарт ISO/IEC 19501. В 2012 г. UML версии 2.4.1 принят как стандарт ISO/IEC 19505-1 и 19505-2.

Известны предметно-ориентированные разработки для алгоритмизации задач в сфере медицины. В первую очередь это Arden Syntax [4] — язык разметки для представления и распространения медицинских знаний. Спецификация Arden Syntax является частью международного медицинского стандарта Health Level 7 (HL7) и используется для кодирования знаний, представленных в медицинских логических модулях (MLM). Для клинических руководств разработан компьютерный формат Guideline Interchange Format (GLIF), который предназначен для моделирования и совместного использования клинических руководств среди медицинских учреждений. GLIF основан на Arden Syntax. С 2000 г. используется третья версия — GLIF3 [5].

Следует упомянуть также проект визуального языка «ДРАКОН», который предлагается использовать и для записи медицинских алгоритмов [6].

Существующие инструментальные средства верификации, которые позволяют оценивать корректность схем алгоритмов, ориентированы в первую очередь на работу с UML-диаграммами. Сравнительный анализ ряда таких инструментов дан, например, в [7]. Короткий список типичных ошибок, которые можно выявить при анализе UML-диаграмм приведен в [8].

Модельно ориентированный подход для верификации клинических руководств, представленных в виде диаграмм состояний UML дан в [9]. Метод проверки клинических руководств на соответствие требованиям, определенным на основе набора шаблонов свойств спецификации, разработан в рамках проекта GLARE (GuideLine Acquisition, Representation and Execution) в [10].

Схемы алгоритмов, которые приведены в утвержденных МОЗ Украины документах, наиболее близки к блок-схемам действующего ГОСТ 19.701-90. Авторы статьи полагают, что в настоящее время переходить на какой-либо другой стандарт нецелесообразно и преждевременно. Необходимо выявить несоответствия этому стандарту в уже разработанных документах и внести соответствующие исправления в новых версиях документов по стандартизации медицинской помощи. Их пересмотр и обновление должны выполняться каждые пять лет, согласно утвержденному графику.

3. Объект, цель и задачи исследования

Объектом исследования являются схемы алгоритмов диагностики и лечения из медико-технологических документов по стандартизации медицинской помощи, утвержденных и внедряемых согласно приказам МОЗ Украины (унифицированные клинические протоколы медицинской помощи и адаптированные клинические руководства, основанные на доказательствах).

Цель исследования — определить способ для выявления несоответствий (ошибок) в указанных схемах алгоритмов действующему стандарту.

Для достижения поставленной цели необходимо выполнить такие задачи:

1. Определить наиболее характерные несоответствия (ошибки) в схемах алгоритмов.
2. Составить алгоритм для нахождения несоответствий (ошибок) в схемах алгоритмов.
3. Проверить адекватность созданного алгоритма.

4. Алгоритм проверки схем медицинских алгоритмов

Эмпирический обзор схем медицинских алгоритмов, включенных в утвержденные медико-технологические документы дает нам такой предварительный список типичных несоответствий: отсутствия символа решения (логического блока, ромба); отсутствие терминаторов (начало и конец алгоритма); отсутствие линий («висящие» блоки); нестандартные символы (блоки); неоднозначные описания условий в символах решения.

Следовательно, для проведения детального анализа необходимо в первую очередь выявлять именно такие несоответствия. Для этого необходимо задать формализованный метод и описать последовательность (алгоритм) действий.

ISSN 2226-3780

Для выявления несоответствий в схемах алгоритмов авторами статьи был подготовлен метод проверки схем на соответствие требованиям стандарта ГОСТ 19.701-90. Для записи большинства рассматриваемых алгоритмов достаточно использовать подмножество символов, определенных в стандарте. Это подмножество состоит из таких трех символов (блоков): процесс, решение и терминатор. Символы различаются по трем признакам: графическое начертание, количество входов и выходов, а также использованием характерных слов, которые употребляются в тексте записей внутри символов. Любое расхождение между этими наблюдаемыми признаками указывает на возможность наличия ошибки (несоответствия стандарту) в этом символе. Фактически здесь следует поочередно проверить каждый блок на совпадение указанных признаков.

Алгоритм проверки блок-схемы медицинского алгоритма (псевдокод):

— Начало алгоритма.

— Пронумеровать все блоки (символы процесса и специальные символы).

— Начало цикла по блокам. Для каждого блока (символа процесса или специального символа) выполнить:

1) определить тип блока (символ: процесс, решение, терминатор, комментарий, другое).

ЕСЛИ это нестандартный символ, ТО отметить наличие ошибки;

2) Определить линии входов и выходов. Подсчитать количество входов и выходов;

3) ЕСЛИ это символ процесса (прямоугольник) ТО ЕСЛИ NOT ((текст внутри символа или его комментария описывает процесс) AND (в тексте нет слов, описывающих решение: «если», когда, в случае, при условии и т. п.) AND (в тексте нет слов, описывающих цикл: для всех, с каждым, по очереди и т. п.) AND (количество выходов = 1)) ТО отметить наличие ошибки.

4) ЕСЛИ это символ решения (ромб) ТО ЕСЛИ NOT ((в тексте определено условие) AND (текст не описывает процесс) AND (в тексте нет слов, задающих циклический оператор) AND (количество выходов >= 2) AND (для каждого альтернативного выхода есть подпись — значение условия)) ТО отметить наличие ошибки;

5) ЕСЛИ NOT ((это символ терминатора) AND ((количество входов = 0) OR (количество выходов = 0))) ТО отметить наличие ошибки.

— Конец цикла по блокам.

— Подсчитать количество символов (блоков), у которых количество входов = 0.

ЕСЛИ NOT (количество таких блоков = 1) ТО отметить наличие ошибки в алгоритме.

— Подсчитать количество блоков (символов), у которых количество выходов = 0.

ЕСЛИ (количество таких блоков < 1) ТО отметить наличие ошибки в алгоритме.

— Проверить каждый блок, в котором подозреваем наличие ошибки. Подсчитать количество блоков, в которых найдены ошибки. Определить тип ошибки. Подсчитать количество ошибок разных типов.

— Конец алгоритма.

При выполнении этого алгоритма необходимо пронумеровать все символы (блоки) исследуемой схемы. Нумерация выполняется вручную в графическом редакторе или в редакторе схем, в котором открываем документ, содержащий исследуемую схему. Желательно выполнять нумерацию вдоль направления потока и в стандартном направлении (слева направо и сверху вниз).

На рис. 1 дана схема исследуемого в статье алгоритма проверки схем алгоритма требованиям ГОСТ. Результаты проверки занесены в таблицу для дальнейшего анализа.

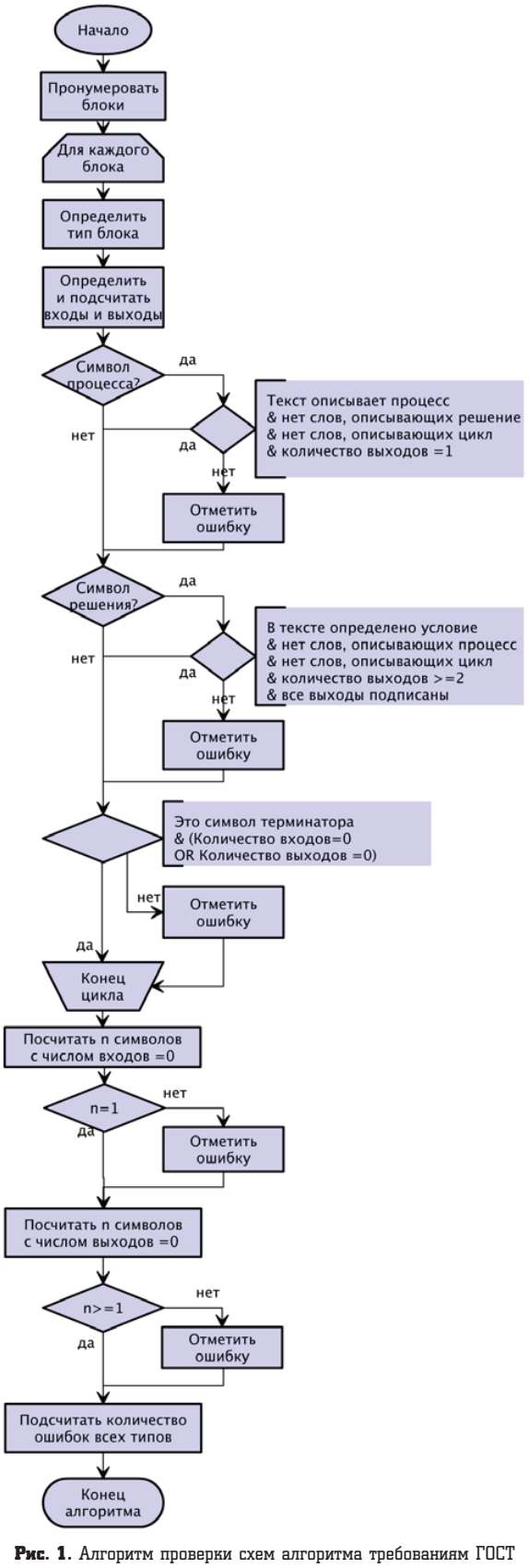

Рис. 1. Алгоритм проверки схем алгоритма требованиям ГОСТ

Алгоритм не предназначен для исправления схем. Задача статьи — выявление несоответствий в схеме. Авторы статьи также не занимались анализом полного текста алгоритма, который приведен в текстах медико-технологических документов.

5. Результаты применения алгоритма

В 2015/2016 учебном году несколько студентов второго курса Днепропетровской медицинской академии, изучающих дисциплину «Медицинская информатика», получили задание выполнить по авторской методике проверку схем алгоритмов из утвержденных медико-технологических документов. Эти задания выполнялись в рамках индивидуальной самостоятельной работы, которая предусмотрена в учебном плане дисциплины. Постановка задачи по проверке алгоритма обсуждалась авторами статьи в [11].

На рис. 2 представлен фрагмент схемы алгоритма из адаптированного клинического руководства для лечения псориаза [12] после нумерации блоков.

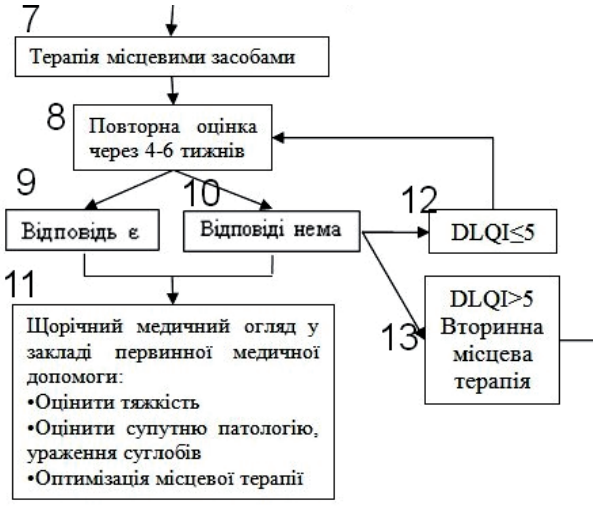

Рис. 2. Фрагмент алгоритма лечения и ухода за пациентами с псориазом и псориатическим артритом [12]

Результаты выполненной проверки записываем в таблицу для дальнейшего подсчета частот встречаемости различных ошибок. В табл. 1 показаны записи, сделанные для данного фрагмента алгоритма.

Видно, что проверка прошла успешно только для одного символа — блока № 7. Все остальные блоки в данном фрагменте схемы алгоритма получили отметку о наличии ошибки, которая потом была определена как одна из типичных ошибок. Для некоторых блоков отмечены несколько ошибок. Чаще всего (4 раза) наблюдаем здесь ошибку, когда значение условия записано как процесс, т. е. вместо надписи возле линии текст помещен в прямоугольник. Дважды отмечена ошибка, когда блок решения совмещен с процессом и показан прямоугольником без записи условия ветвления внутри.

В схемах медицинских алгоритмов в других исследованных документах, выбранных авторами статьи для проверки, также были обнаружены ошибки. Более того, несоответствия стандарту были найдены во всех проверенных схемах алгоритмов.

6. Обсуждение первых результатов использования разработанного метода проверки схем медицинских алгоритмов

Выполненная формализованная проверка подтвердила системный характер несоответствий схем рассматриваемых медицинских алгоритмов действующему ГОСТу. В большинстве случаев значительное количество блоков схемы требуют исправлений.

Классификация выявляемых несоответствий (ошибок) позволяет отнести практически все из них к одному из предварительно определенных типов ошибок схем алгоритмов. Далее для каждого такого типа ошибок будут даны рекомендации по изменению схемы алгоритма для их исправления.

Подготовленный авторами статьи алгоритм проверки не требует высокой квалификации исполнителя. Функционально грамотному исполнителю достаточно лишь освежить свои знания по теме «алгоритмы» из курса медицинской информатики.

Исполнителем данного в работе алгоритма является человек. Автоматизация исполнения не целесообразна, по той причине, что затраты на приведение схем медицинских алгоритмов к машинно-читаемому виду вероятно будут большими из-за непосредственной проверки и исправления схем этих алгоритмов руками человека. Задачу автоматизации проверки целесообразно рассматривать лишь, когда количество подлежащих проверке медицинских стандартов и унифицированных клинических протоколов будет исчисляться тысячами документов.

Таблица 1

Результаты проверки соответствия фрагмента схемы алгоритма

№ блока	Входов	Выходов усл. 4 (=1)	Тип блока	Условие 1 (процесс)	Условие 2 (не решение)	Условие 3 (не цикл)	Код типичной ошибки
7	1	1	процесс	true	true	true	—
8	2	2	процесс	true	true	true	3. Процесс вместо решения 5. Совмещение действий
9	1	1	процесс	false	—	—	4. Значение записано как процесс
10	1	3	процесс	false	—	—	3. Процесс вместо решения 4. Значение записано как процесс 9. Иная ошибка
11	1	0	процесс	true	true	true	2. Нет терминатора «конец» 6. Нет выхода
12	1	1	процесс	false	—	—	4. Значение записано как процесс
13	1	1	процесс	true	false	true	4. Значение записано как процесс

В дальнейшей авторы статьи намерены найти все несоответствия указанному ГОСТу в схемах алгоритмов из унифицированных клинических протоколов, принятых в последние годы, которые доступны в реестре медико-технологических документов по стандартизации медицинской помощи.

Следующим этапом работы авторов статьи будет разработка метода для исправления типичных ошибок (несоответствий ГОСТ), которые могут быть выявлены данным алгоритмом.

7. Выводы

В результате проведенных исследований:

1. Представлен разработанный алгоритм для нахождения несоответствий (ошибок) в схемах медицинских алгоритмов.

2. Выполнена проверка созданного алгоритма на реальных исходных данных. Подтверждена возможность его применения исполнителями, не являющимися специалистами в области информационных технологий.

3. Подтверждена эмпирическая оценка частоты обнаружения несоответствий. Подтверждено предположение об ограниченности набора типичных несоответствий.

Литература

1. Ярош, Н. П. Сучасний стан, проблеми стандартизації медичної допомоги та шляхи їх вирішення в умовах реформування системи охорони здоров'я України [Текст] / Н. П. Ярош, С. І. Лупей-Ткач // Україна. Здоров'я нації. — 2012. — № 1. — С. 95–100.
2. Бліхар, В. Є. Аналіз медико-технологічних документів зі стандартизації медичної допомоги та обґрунтування шляхів їх удосконалення [Текст] / В. Є. Бліхар та ін. // Вісник соціальної гігієни та організації охорони здоров'я України. — 2010. — № 4. — С. 72–80.
3. Дубинский, А. Г. Графическая визуализация медицинских алгоритмов диагностики и лечения [Текст] / А. Г. Дубинский, О. А. Хорольский // 15 Міжнародна науково-технічна конференція SAIT 2013 «Системний аналіз та інформаційні технології». — К.: ННК «ІПСА» НТУУ «КПІ», 2013.
4. Samwald, M. The Arden Syntax standard for clinical decision support: Experiences and directions [Text] / M. Samwald, K. Fehre, J. de Bruin, K.-P. Adlassnig // Journal of Biomedical Informatics. — 2012. — Vol. 45, № 4. — P. 711–718. doi:10.1016/j.jbi.2012.02.001
5. Peleg, M. GLIF3: the evolution of a guideline representation format [Text] / M. Peleg et al. // Proc. of the AMIA Symposium. — American Medical Informatics Association, 2000. — P. 645–649.
6. Пароннджанов, В. Д. Зачем врачу блок-схемы алгоритмов? Удобный способ представления медицинских знаний и предотвращения врачебных ошибок. Алгоритмы для эффективного клинического мышления [Электронный ресурс] / В. Д. Пароннджанов. — М.: Препринт, 2016. — 219 с. — Режим доступа: \www/URL: http://drakon.su/_media/16._zachem_vrachu_2016_17_glav_127_risunkov_219_str_literat_200.pdf
7. Литвинов, В. В. Инструментальные средства верификации моделей программного обеспечения [Текст] / В. В. Литвинов, И. В. Богдан, К. С. Сливко // Вісник Чернігівського державного технологічного університету. Серія: Технічні науки. — 2013. — № 2. — С. 120–125.
8. Бузовский, О. В. Система верификации графических схем алгоритмов и генерации программных кодов [Текст] / О. В. Бузовский, А. В. Алещенко // Проблеми інформатизації та управління. — 2015. — Т. 2, № 50. — С. 32–35.
9. Pérez, B. Authoring and verification of clinical guidelines: A model driven approach [Text] / B. Pérez, I. Porres // Journal of Biomedical Informatics. — 2010. — Vol. 43, № 4. — P. 520–536. doi:10.1016/j.jbi.2010.02.009
10. Bottrighi, A. Adopting model checking techniques for clinical guidelines verification [Text] / A. Bottrighi, L. Giordano, G. Molino, S. Montani, P. Terenziani, M. Torchio // Artificial Intelligence in Medicine. — 2010. — Vol. 48, № 1. — P. 1–19. doi:10.1016/j.artmed.2009.09.003
11. Дубинский, А. Г. Метод проверки блок-схем медицинских алгоритмов [Электронный ресурс] / А. Г. Дубинский, О. А. Хорольский // Материалы международной научно-технической конференции «Информационные технологии в металлургии и машиностроении». — Днепропетровск: НМетАУ, 2016. — Режим доступа: \www/URL: http://nmetau.edu.ua/file/itmm_2016_program.pdf
12. Адаптированное клиническое руководство, основанное на доказательствах «Псориаз, включая псориатические артропатии» [Электронный ресурс]: Приказ МОЗ Украины от 20.11.2015 № 762. — Режим доступа: \www/ URL: http://www.dec.gov.ua/mtd/reestr_r.html

РОЗРОБКА МЕТОДА ПЕРЕВІРКИ БЛОК-СХЕМ МЕДИЧНИХ АЛГОРИТМІВ

Забезпечення високої якості медичної допомоги вимагає точного та ефективного запису медичних алгоритмів. Запропоновано метод для виявлення частин схем алгоритмів, які записано не відповідно до державного стандарту, що допускає їх неоднозначне тлумачення та може стати чинником появлення лікарських помилок. Наведено алгоритм знаходження таких частин, надано приклад його застосування для знаходження помилок.

Ключові слова: алгоритм, стандарт, блок-схема, клінічна настанова, медичний стандарт, клінічний протокол.

Дубинский Алексей Георгиевич, *кандидат технических наук, доцент, кафедра медико-биологической физики и информатики, Днепропетровская медицинская академия, Украина,* ***e-mail: dubinsky@ukr.net****.*

Хорольский Олег Алексеевич, *лаборант, кафедра биохимии, медицинской и фармацевтической химии, Днепропетровская медицинская академия, Украина.*

Дубінський Олексій Георгійович, *кандидат технічних наук, доцент, кафедра медико-біологічної фізики та інформатики, Дніпропетровська медична академія, Україна.*

Хорольський Олег Олексійович, *лаборант, кафедра біохімії, медичної та фармацевтичної хімії, Дніпропетровська медична академія, Україна.*

Dubinsky Alexey, *State Establishment «Dnipropetrovsk Medical Academy», Ukraine,* ***e-mail: dubinsky@ukr.net****.*

Khorolsky Oleg, *State Establishment «Dnipropetrovsk Medical Academy», Ukraine*