

УДК 681.3.06

Бикбаев Дмитрий Ринадович

Киркоров Сергей Иванович

Обработка текста на языке Ada

Компактные программы и целые системы автоматической обработки неструктурированных текстов – одно из важнейших направлений современных информационных технологий. Их многообразие и сложность задач, которые они решают, позволяет выделить их и провести систематизацию и классификацию с целью упрощения выбора решения, наиболее адекватного для конкретной задачи [1].

По классификации [1], обработку неструктурированной документации, историй болезни, патентов и диссертаций и т. п. – можно разбить на две условные категории.

Задачи, с которыми ежедневно сталкивается любой пользователь – проверка орфографии, фильтрация спама, автоматический перевод небольших фрагментов текста (несколько предложений) и др., которые в области автоматической обработки текстов почти решены. Имеются доступные программные средства в виде библиотек или отдельных модулей, из которых можно выстроить последовательность автоматической обработки.

Ко второй категории относятся задачи "требующие обработки больших текстовых массивов: нахождение релевантных ответов на вопросы (задачи «вопрос-ответ»), полноценный машинный перевод целостных текстов, анализ мнений и отзывов, конструирование рекомендательных систем, работающих с большими массивами неструктурированных данных. Отличительная особенность таких задач – их сложность и отсутствие формализации, приводящие к тому, что для них пока еще нет полноценного набора решений, а применяются вспомогательные методы выделения ключевых слов и словосочетаний, суммаризации (автоматического реферирования) текстов и классификации текстов" [1]. Компьютерная лингвистика обеспечивает методами решения задач второй категории, например: методы машинного обучения, статистического анализа, модели Маркова, логические модели и модификации этих методов с учетом специфики Больших Данных [2].

Задача автоматизации поиска причин отказов и сбоев в работе систем имеющих несколько уровней структурной иерархии, где каждая подсистема является объектом содержащим связанные между собой протоколом взаимодействия узлы и подсистемы можно отнести ко второй категории задач. Решение их выполняется путем анализа отчетов (так называемых лог-файлов) протоколов взаимодействия подсистем, выполненных тестов на различных уровнях состояния системы и подсистем. Полная автоматизация анализа отчетов в общем случае не решена. Процесс поиска причин отказов и сбоев – итерационный и количество итераций в значительной степени зависит от эксперта: его знаний и опыта.

Примером такой системы может быть совокупность информационных технологий применяемые в банках и в частности подсистема, включающая в себя банкоматы (АТМ). Сам банкомат – критически важный объект информационной инфраструктуры. Климатические условия, в которых он должен функционировать, удаленность банкомата от управляющего им хоста и часто не гарантированным уровнем сервиса каналов связи, низкий технический уровень потребителя банковских услуг приводит к возникновению отказов и сбоям в работе банкомата. Простой в нерабочем состоянии или сбой его работы может привести к финансовым потерям, которые нельзя игнорировать. Вопрос автоматизации поиска причин и сбоев в работе банкоматов и сокращение времени и финансовых затрат на восстановления их работоспособности остается актуальным и в настоящее время.

Программное обеспечение, поставляемое производителем банкоматов, имеет соответствующее заключение на сертификацию по PCI DSS и обеспечивает систему тестирования и протоколирования работы банкомата в соответствующих текстовых файлах с детализацией до узла который можно идентифицировать и заменить/отремонтировать. Таким образом, объем текста, который необходимо изучить и проанализировать инженеру эксперту может достигать размеров десятков и в некоторых случаях сотни мегабайт. Так как в каждом конкретном банке имеются свои особенности в аппаратной и программной части, не

Обработка текста на языке Ada

УДК 681.3.06 Бикбаев Д. Р. Киркоров С. И.

существует универсального программного обеспечения решающего задачу в полном объеме автоматизации работы инженера эксперта. Был создан, расширяемый комплект средств автоматизации инженера эксперта для анализа файлов протоколов работы банкомата. При создании средств автоматизации были выполнены следующие требования:

1. Автоматизировать опыт работы накопленный в течении более 10 лет более трех банков Республики Беларусь с сетью банкоматов различных поколений с программным обеспечением компании NCR.
2. Не иметь побочных эффектов, в том числе не требовать предустановки в операционной системе.
3. Не оказывать влияние на сертифицированное по PCI DSS программное обеспечение.
4. Средства разработки должны иметь стандартизированные средства/библиотеки обработки структурированной и не структурированной текстовой информации.
5. Понятный и удобный в сопровождении исходный код программ.
6. Возможность переноса исходного текста программ под разные операционные системы.
7. Экономическая эффективность разработки и использования комплекса средств автоматизации.

Были рассмотрены средства разработки для языков программирования: Ada, Python, Java, C++, C# и Visual Basic. Действительно, все эти языки упрощают писать код, вести разработку быстро и очень гибко, предлагают функции с динамическим поведением, и некоторые даже позволяют задавать компиляторам вероятное намерение разработчика для генерации кода проверки заданных ограничений во время исполнения кода.

Разработанные для общего рынка программирования, языки Python, Java, C++, C# и Visual Basic стали плохо удовлетворять требованиям для работы в домене высоко доверительных систем сегментом которых являются банковские. В информационной структуре с высокими требованиями к надежности, безопасности и целостности, такие которые предъявляются при производстве банкоматов, и в любом устройстве (для самолетов, ракет, спутников, поездов), чей отказ мог подвергнуть опасности критические активы или человеческую жизнь, используются другие языки программирования. Эти языки программирования должны поддерживать верхний уровень стандарта разработки программного обеспечения, необходимый, чтобы поддержать обеспечение целостности и безопасности системы.

Понятие проверки – практики показа, что система ведет себя и выступает, как предназначено – является ключом для доказательства протокольной реализации в среде критически важных объектов информационной инфраструктуры. Проверка может быть выполнена некоторой комбинацией анализа, тестирования, статического анализа, и применением формальных методов доказательства. Обеспечение высокого уровня доверия к программному обеспечению и увеличивающаяся сложность современных систем сделали это более трудной задачей. Технологии и методы, которые, возможно, были совершенно приемлемы десять или пятнадцать лет назад, недостаточны сегодня. К счастью состояние в анализе и инструментах доказательства и методах также совершенствовалось.

Последние версии языка Ada, Ada 2005 и Ada 2012, делают возможной обеспечить высокий уровень доверия к программному обеспечению. От ее начала в 1980-х, когда язык программирования Ada был разработан, чтобы удовлетворить требования систем высокого уровня доверия, и продолжает быть таковым для реализации критических важных объектов информатизации встроенных или собственных программных приложений. И это получило недавно повышенное внимание. Каждая версия языка улучшила выразительность во многих областях. В языке программирования Ada 2012, в частности, имеет новый функционал, который основан на таком понятие как контракт программирования. Эти новые возможности ценны к любому проекту, где проверка – часть технического жизненного цикла программного обеспечения. Вместе с этими улучшениями языка, компилятором Ada и технологическими инструментами также сохранилась совместимость с общими вычислительными разработками за прошлые несколько лет. Среды разработки Ады доступны на широкий диапазон платформ и используется для самых требовательных приложений.

Обработка текста на языке Ada УДК 681.3.06 Бикбаев Д. Р. Киркоров С. И.

Не будем утверждать, что Ada всегда – универсальное решение, лучший выбор, чем какой-либо другой язык; Ada больше не серебряная пуля. В некоторых областях информационных технологий применение других языков экономически целесообразно из-за доступности определенных библиотек или платформ разработки. Например, считается хорошим выбором C++ и Java для настольных программ или для приложений, где сокращенное время выхода на рынок – главная цель. В другой области приложений, такой как программирование веб-сайта или системное администрирование, предпочтение отдают другим формализмам, таким как скрипт сценария и интерпретируемые языки. Чтобы выполнить требования, предъявляемые к программным средствам, ключевым моментом является выбор надлежащего технического подхода, с точки зрения языка и инструментария разработки программного обеспечения. Сила языка программирования Ada проявляется в областях, где обеспечение надежности и безопасности с высоким уровнем доверия к программным средствам является главной задачей.

Опыт реализации комплекта программ, который автоматизирует работу инженера эксперта, для анализа файлов протоколов работы банкомата, доказал правильность выбора языка программирования Ada-2012.

ЛИТЕРАТУРА

1. Дмитрий Ильвовский, Екатерина Черняк, Системы автоматической обработки текстов // Открытые системы. – 2014. – № 01. – Электронный ресурс <http://www.osp.ru/os/2014/01/13039687/>
2. Sergei O. Kuznetsov, Fitting Pattern Structures to Knowledge Discovery in Big Data. ICFCFA 2013. P. 254–266

БИКБАЕВ Дмитрий Ринадович – ведущий инженер-программист ООО Картсервис г. Минск, Республика Беларусь.

Научные интересы:

информационные технологии, автоматизация тестирования и сертификации программного обеспечения.

КИРКОРОВ Сергей Иванович – ведущий инженер-программист ООО Картсервис г. Минск, Республика Беларусь. Руководитель НПП МедиаСкан.

Научные интересы:

информационные технологии, исследование языков программирования, автоматизация тестирования и сертификации программного обеспечения.