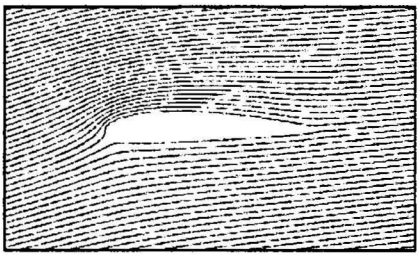

Рис. 1

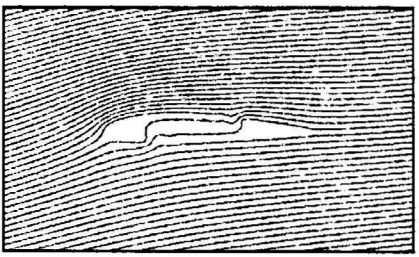

Рис. 2

Работа выполнена в рамках проектов №2.1.1/3828 и №2.1.1/12925 конкурсной программы "Развитие научного потенциала высшей школы (2009 – 2011 гг.)" Минобрнауки Российской Федерации.

ЛИТЕРАТУРА

1. Лежнев А. В., Лежнев В. Г. *Метод базисных потенциалов в задачах математической физики и гидродинамики.* – Краснодар: Кубанский гос. ун-т, 2009. – 134 с.

2. Jacobs E. N., Ward K. E., Pinkerton R. M. *The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel.* – T.R. No 460 Washington, D.C.: N.A.C.A., 1930. – 61 p.

3. Владимиров В. С. *Уравнения математической физики.* – М.: Наука, 1981.

**А. С. Тощев**

*Казанский (Приволжский) федеральный университет,*
*sanchis.no@gmail.com*

## К НОВОЙ КОНЦЕПЦИИ АВТОМАТИЗАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Существует множество инструментов для автоматизации разработки, например, такие современные интегрированные

среды разработки, как Microsoft Visual Studio 2010 [1], Microsoft Visual Studio 2008 [1], IntelliJ IDEA [2], включают технологии типа IntelliSense [3] (автодополнения ввода пользователя). Стоит отметить и богатые средства моделирования UML [4], которые включают множество готовых архитектурных шаблонов, а также средства управления циклом разработки, такие, например, как MAVEN [5] (сборщик приложений), который включает множество шаблонов для генерации скелета приложения. Основной минус этих систем состоит в том, что они не "понимают" архитектуры приложения как целостной системы. Например, возможна ситуация, когда инструмент генерирует код приложения, разработчик заполняет недостающие тела методов, и возникает необходимость добавления полей в базу данных. Инструмент опять генерирует код, а все приложения и результаты деятельности разработчика оказываются потерянными. В целом все известные инструменты автоматизации программного обеспечения автоматизируют не весь цикл разработки, а только его части.

В 2005 году Массачусетским технологическим институтом опубликовано исследование по генерации классов Python на основе сокращенного английского языка [6]. Подход заключался в разборе текста на естественном языке и переводе его в семантическую модель концепций языка программирования.

Наше представление об эффективной автоматизации разработки программного обеспечения заключается в следующем: входными параметрами системы являются бизнес-требования, созданные аналитиком; выходным параметром является готовый код приложения. Бизнес-требования должны отвечать следующим условиям: отсутствие неоднозначности; грамматическая правильность; отсутствие противоречий (должно быть

выявлено системой и подтверждено экспертом).

Изначальная идея была в том, чтобы улучшить прототип Metafor Массачусетского технологического института и создать "понимательный" механизм архитектуры приложения. Выяснилось, что прямолинейное сопоставление объектов текста и классов приложения проблему не решает, поэтому мы добавили промежуточную метамодель требований: был использован уже готовый лексический обработчик Stanford Parser [7]. Отталкиваясь от него, мы формализовали архитектуру генератора: Лексический обработчик — Понимательный модуль — Модуль генерации решения. Все модули используют базу знаний для хранения информации. В качестве дополнения может быть создан модуль общения, с помощью которого остальные модули смогут общаться с экспертом.

Данная система является Open Source проектом Menta [8]. По нашим подсчетам, в конечном итоге нам удастся автоматизировать как минимум 10 % всех задач разработки, то есть убрать тривиальные задачи и дать возможность разработчику решать сложные и интересные задачи.

## ЛИТЕРАТУРА

1. *Visual Studio* //Wikipedia. – http://ru.wikipedia.org/wiki/Visual_Studio.

2. *IntelliSense* //Wikipedia. – http://ru.wikipedia.org/wiki/IntelliJ_IDEA.

3. *IntelliJ IDEA* //Wikipedia. – http://ru.wikipedia.org/wiki/IntelliSense.

4. *UML* //Wikipedia. – http://ru.wikipedia.org/wiki/UML.

5. *Introduction to the build lifecycle* //Apache Maven Project, Apache Software foundation. – http://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html.

6. Lieberman H., Liu H. *Metafor: visualizing stories as code* // Proc. of the ACM Int. Conf. on Intelligent User Interfaces IUI-2005, Jan. 9 – 12, 2005, San Diego, CA, USA. – ACM Press, 2005. – P. 305-307. – http://larifari.org/writing/IUI2005-Metafor.pdf.

7. Klein D., Manning C. *The Stanford parser: a statistical parser* // The Stanford Natural Language Processing Group. – http://nlp.stanford.edu/software/lex-parser.shtml.

8. *The Menta Project.* – http://menta-project.org.

**П. И. Трошин**

*Казанский (Приволжский) федеральный университет,*
*Paul.Troshin@gmail.com*

## РАЗРАБОТКА АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ СТУДЕНТОВ ПО МАТЕМАТИКЕ НА ПРИМЕРЕ ГЕОМЕТРИЧЕСКИХ ЗАДАЧ

При проведении практических занятий по математике возникает постоянная необходимость в составлении новых контрольных работ и промежуточных тестов, причем с большим числом вариантов. Облегчить работу в этом направлении можно двумя способами (а также их совмещением): формировать содержание билета, используя некоторый автоматизированный процесс; выносить промежуточные тестирования в режим онлайн в интернете.

Мы разрабатываем систему автоматического формирования билета на основе связки программ Mathematica и LaTeX, а также систему оплайн-тестирования по этим билетам: