§ 4 МЕТОДЫ, ЯЗЫКИ И МОДЕЛИ ЧЕЛОВЕКО-МАШИННОГО ВЗАИМОДЕЙСТВИЯ

В.В. Килеев

КОМПОНЕНТЫ АРХИТЕКТУРЫ КОМПЬЮТЕРНОЙ СИСТЕМЫ ВЕРИФИКАЦИИ ОРФОГРАФИИ ФИННО-УГОРСКИХ ЯЗЫКОВ

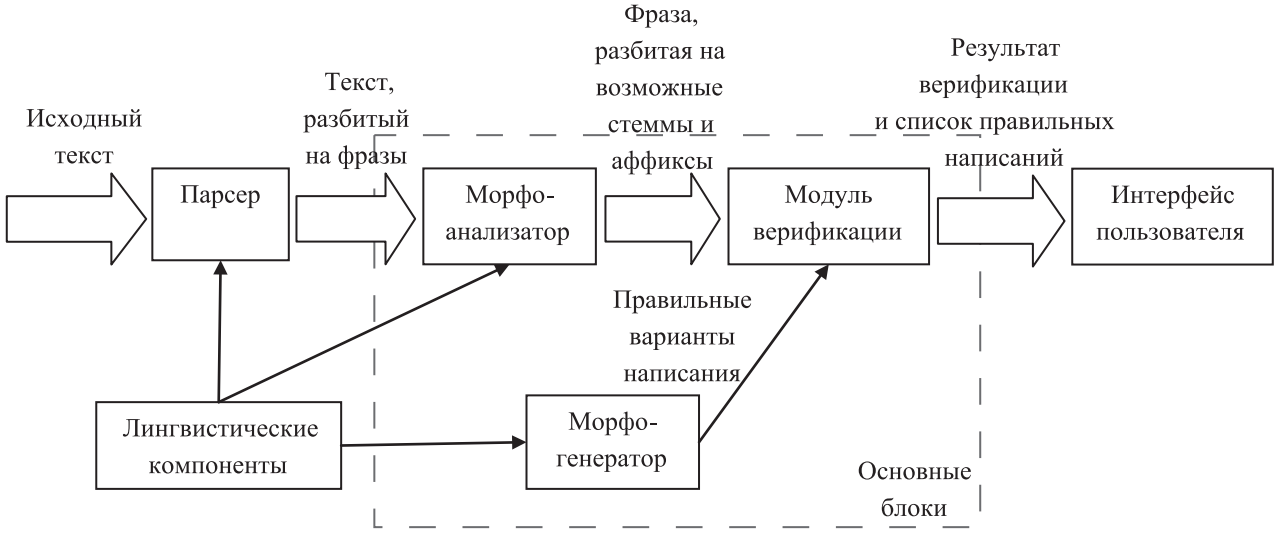
Аннотация: *Целью работы является рассмотрение архитектуры разработанной системы верификации орфографии финно-угорских языков. Архитектура разработанной системы разбита на функциональные блоки. Каждому функциональному блоку дается дается подробное описание. На приведенном рисунке рассматривается взаимоотношение функциональных блоков в системе. В работе также рассматриваются основные преимущества разработанной системы перед существующими и дается диаграмма использования разработанной системы.*

Ключевые слова: *программное обеспечение, верификация орфографии, проверка орфографии, компьютерная лингвистика, обработка естественных языков, архитектура системы, финно-угорские языки, спелл-чекер, функциональные блоки, лингвистические компоненты.*

Актуальность рассмотрения архитектуры системы верификации орфографии объясняется необходимостью использования унифицированного подхода к решению вопроса верификации орфографии, с разграничением функционирования лингвистических компонентов языка.

Архитектура известных систем использует информационную базу [1]. Предложена архитектура, компоненты которой взаимодействуют согласно схеме, представленной на рис. 1, которая использует для хранения и манипулирования лингвистическими компонентами базу данных. Преимуществом использования базы данных перед информационной базой состоит в обеспечении возможности и пользователю-лингвисту и разработчику программного обеспечения системы полноценно работать с лингвистическими компонентами языка, манипулировать информацией и компонентами данных.

Рис. 1. Схема взаимодействия компонентов архитектуры системы верификации орфографии



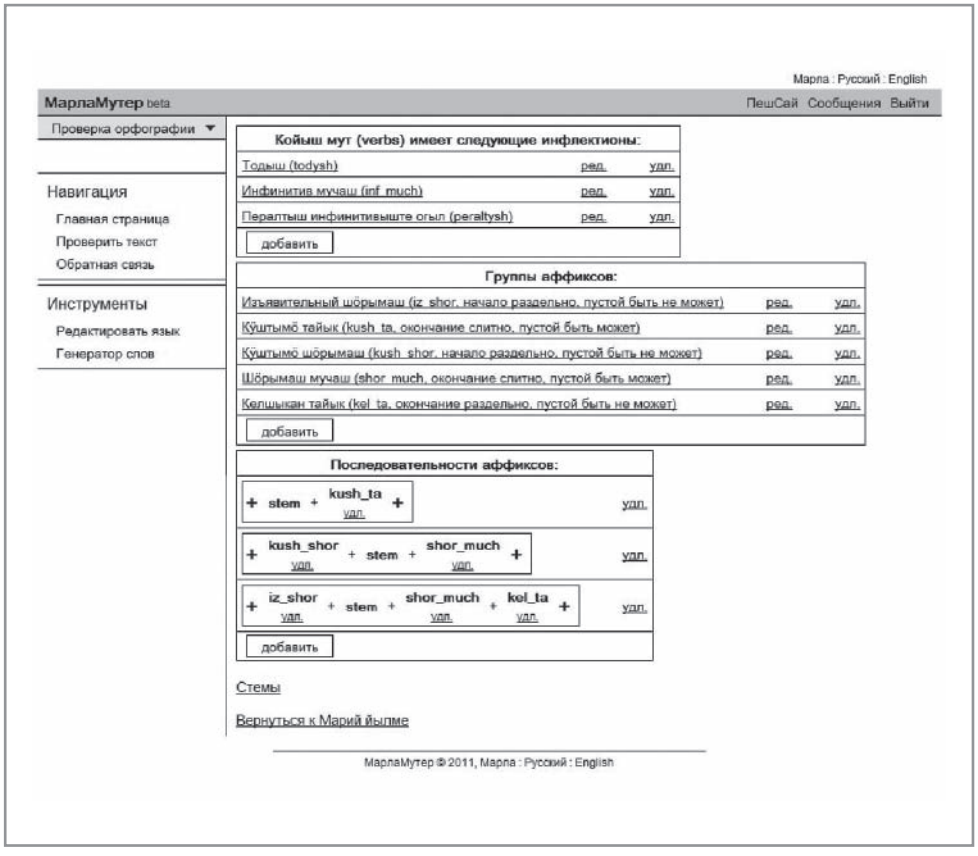
Предложенная архитектура, разрабатываемой системы[2], представляется следующими функциональными блоками:

- Функциональный блок парсинга, основное назначение которого – это разбивка текста на фразы. Он использует лингвистические компоненты для получения информации о возможных аффиксах, которые пишутся раздельно. В обычных системах верификации орфографии, текст разбивается только на отдельные слова, что не дает возможности обрабатывать такие лингвистические конструкции как «ом кошт ыле» – глагол с отрицательной частицей и частицей условного наклонения. Благодаря тому, что частицы рассматриваются парсером как аффиксы и в лингвистическом блоке хранится информация об отдельно пишущихся аффиксах, это дает возможность уловить ошибки неправильного использования этих частиц модулем верификации. Это составляет отличие предлагаемого блока парсинга от парсинга, осуществляемого в существующих системах.
- Морфоанализатор, который используется для анализа каждой фразы. На выходе морфоанализатора получается список с возможной разбивкой фразы на стеммы и аффиксы. Он работает по разработанному унифицированному алгоритму, т.е. не привязан к какому-то конкретному языку. Все языковые данные получаются морфоанализатором из лингвистических компонент, составленных для исследуемого языка. В блоке морфоанализатора используется алгоритм стемминга[3], отличающийся от существующих[4][5][6] возможностью работать с аффиксами, пишущимися раздельно и длинными цепочками аффиксов, состоящими (теоретически), из неограниченного количества элементов.
- Морфогенератор, назначение, которого заключается в генерации всех правильных написаний для заданных стеммов на основе лингвистических компонент.
- Модуль верификации сравнивает результаты, полученные морфоанализатором и сгенерированные морфогенератором. А также составляет список наиболее подходящих написаний, в случае, если верификация не прошла успешно. Для подбора списка наиболее подходящих правильных написаний, предлагаемых пользователю, используется алгоритм расстояния Дамерау-Левенштейна[7].
- Блок лингвистических компонент содержит все лингвистические правила, конструкции и данные, необходимые для осуществления верификации орфографии на языке. Функции данного блока заключаются в представлении всех лингвистических данных языка, необходимых для верификации орфографии на этом языке. Выделение лингвистических данных в отдельный блок позволяет осуществлять верификацию орфографии любого языка, для которого созданы все необходимые лингвистические компоненты. И делает систему в целом независимой от конкретного языка.
- Интерфейс пользователя осуществляет отображение результата верификации пользователю. А также отображает список возможных вариантов замены неправильных словоформ.

Рис 2. ⇨
Пример представления лингвистических компонентов в разрабатываемой системе.

Лингвистические компоненты, входящие в состав разрабатываемого обеспечения системы, можно разбить на следующие категории:

- Инфлектионы, которыми предложено называть грамматические категории, от которых зависит написание части речи. Например, для глаголов в марийском языке могут быть следующие инфлектионы: спряжение, последняя буква корня, ударение не в инфинитиве, окончание инфинитива.
- Допустимые значения инфлектионов – например, для инфлектиона «Спряжение» глаголов марийского языка использованы следующие значения: 1-ое спряжение и 2-ое спряжение.
- Группы аффиксов, где аффиксы объединяются в группы по смысловым значениям, например, группа аффиксов изъявительного наклонения глаголов.
- Аффиксы, разбитые по группам.
- Стеммы – каждому стемму задаются значения инфлектионов.
- Допустимые последовательности групп аффиксов указывают, в какой последовательности могут прикрепляться аффиксы друг к другу или к стемму. Они могут прикрепляться к началу и писаться раздельно через пробел, через дефис или слитно, а также могут прикрепляться к концу и писаться раздельно, через дефис или слитно. Возможность написания аффиксов отдельно через пробел от основной цепочки является особенностью разрабатываемой системы и выделяет ее от существующих систем[8], которые не способны обрабатывать подобные цепочки. Кроме того, возможность составления длинных цепочек аффиксов также является отличием разрабатываемой системы от существующих.



Инфлектионы, значения инфлектионов, группы аффиксов, аффиксы и допустимые последовательности групп аффиксов образуют собой лингвистические правила. Все лингвистические правила поделены на части речи, к которым они относятся. Вместе со стеммами (и в дальнейшем с исключениями) они образуют лингвистические компоненты.

Весь функционал разрабатываемой системы разбит на группы, доступ к которым

распределен между категориями пользователей, как показано на диаграмме вариантов использования (рис. 3). Определим категории пользователей следующим образом.

Категорию пользователей – лингвисты составляют: лингвисты с правами просмотра, лингвисты с правами редактирования и лингвисты с правами администрирования языка.

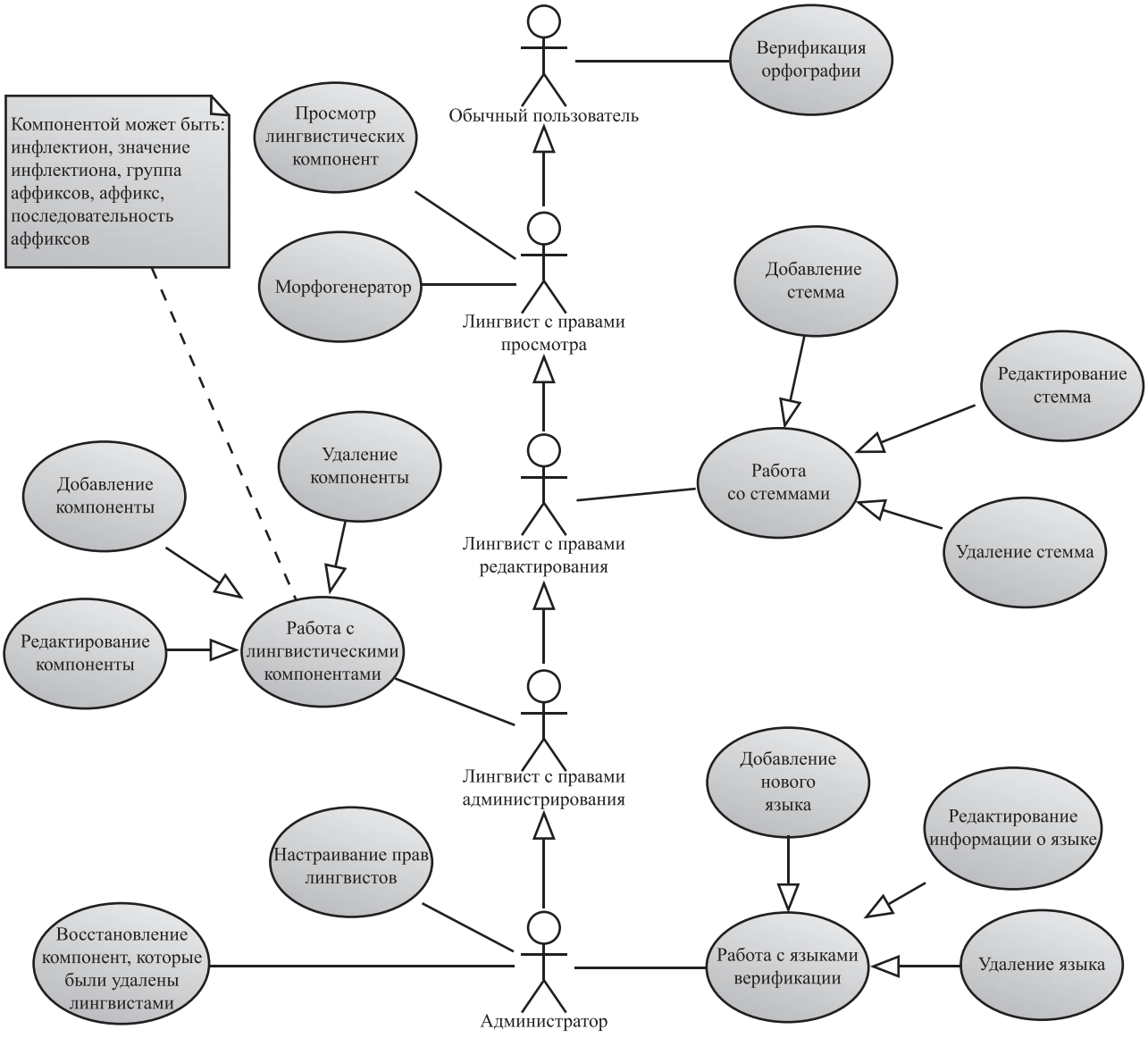


Рис. 3.
Диаграмма вариантов использования разрабатываемой системы

Обычные незарегистрированные пользователи имеют доступ только к непосредственному функционалу верификации орфографии, с которым они взаимодействуют через блок интерфейса пользователя, представленного на рис.1. Их права настраиваются индивидуально для каждого языка. Например, можно иметь права администрирования для удмуртского языка, но права просмотра для марийского языка. Права просмотра дают доступ к просмотру всех

лингвистических компонент языка и доступ к морфогенератору. Права редактирования дают возможность добавлять/редактировать и удалять только стеммы. Права администрирования языка дают возможность добавлять/редактировать и удалять все лингвистические компоненты языка: и стеммы и лингвистические правила.

Категория администратор обладает правами вышеперечисленных и дополнительно возможность настраивать права лингвистов для доступа к тому или иному языку. Ему позволено также, редактировать доступные в системе языки верификации. Кроме того, данная категория пользователей имеет возможность восстанавливать удаленные лингвистами элементы лингвистических компонент.

Таким образом, как отмечалось выше, разработанная архитектура благодаря выделению лингвистических компонент в отдельный независимый блок, позволяет верифицировать орфографию любого языка, для которого будут созданы эти компоненты. При этом, разработанная архитектура дает преимущество перед существующими система, имея возможности работы с аффиксами, пишущимися отдельно от основной словоформы, и возможности работы с длинными последовательностями аффиксов. А возможность разграничения прав доступа и наглядность представления и манипулирования лингвистическими компонентами расширяет круг пользователей, профессиональных лингвистов, использующих компьютерную систему верификации орфографии финно-угорских языков на практике.

Библиография:

1. К.Н. Сануков et al. (ed.) Congressus Decimus Internationalis Fenno-Ugristarum. Йошкар-Ола 15.08.-21.08.2005. Pars IV. Linguistica. Йошкар-Ола: Марийский государственный университет, 2008. Pp. 480-484.
2. Система проверки орфографии МарлаМутер – http://marlamuter.org/checker/
3. Килеев, В.В. Анализ алгоритмов стемминга для формализации компонентов языка финно-угорской группы. / В.В. Килеев, И.Г. Сидоркина // «Труды конгресса по интеллектуальным системам и информационным технологиям «IS&IT'11». Научное издание в 4-х томах» – М.:Физматлит, 2011.-Т3 – С. 47-52
4. J.B. Lovins, 1968: «Development of a stemming algorithm,» Mechanical Translation and Computational Linguistics 11, 22-31.
5. Willett, P. (2006) The Porter stemming algorithm: then and now. Program: electronic library and information systems, 40 (3). pp. 219-223.
6. Harman, D. "How Effective is Suffixing." Journal of the American Society for Information Science 42 (1), 1991, 7-15.
7. Fred J. Damerau, A technique for computer detection and correction of spelling errors, Communications of the ACM, v.7 n.3, p.171-176, March 1964
8. Hunspell: open source spell checking, stemming, morphological analysis and generation under GPL, LGPL or MPL licenses – http://hunspell.sourceforge.net/

References (transliteration):

1. K.N. Sanukov et al. (ed.) Congressus Decimus Internationalis Fenno-Ugristarum. Yoshkar-Ola 15.08.-21.08.2005. Pars IV. Linguistica. Yoshkar-Ola: Mariyskiy gosudarstvennyy universitet, 2008. Pp. 480-484.
2. Sistema proverki orfografii MarlaMuter – http://marlamuter.org/checker/
3. Kileev, V.V. Analiz algoritmov stemminga dlya formalizatsii komponentov yazyka finno-

ugorskoy gruppy. / V.V. Kileev, I.G. Sidorkina // «Trudy kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS&IT'11». Nauchnoe izdanie v 4-kh tomakh» – M.:Fizmatlit, 2011.-T3 – S. 47-52

4. J.B. Lovins, 1968: «Development of a stemming algorithm,» Mechanical Translation and Computational Linguistics 11, 22-31.
5. Willett, P. (2006) The Porter stemming algorithm: then and now. Program: electronic library and information systems, 40 (3). pp. 219-223.
6. Harman, D. "How Effective is Suffixing." Journal of the American Society for Information Science 42 (1), 1991, 7-15.
7. Fred J. Damerau, A technique for computer detection and correction of spelling errors, Communications of the ACM, v.7 n.3, p.171-176, March 1964
8. Hunspell: open source spell checking, stemming, morphological analysis and generation under GPL, LGPL or MPL licenses – http://hunspell.sourceforge.net/