

# РЕДАКТОР ТЕКСТОВ СТРУКТУРИРОВАННОГО ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

Хорошавин В. Д., Насевич П. Е., Садовский М. Е.  
Кафедра интеллектуальных информационных технологий,  
Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь  
E-mail: {vasya.khoroshavin, fallenchromium}@gmail.com, sadovski@bsuir.by

*В данной работе на основании сравнения существующих программных средств для представления и редактирования знаний выдвинуты требования к редактору текстов структурированного представления знаний интеллектуальных систем, а также предложен подход к его построению на основании языка унифицированного смыслового представления знаний и открытой семантической технологии проектирования интеллектуальных систем.*

На сегодняшний день информационные технологии играют неотъемлемую роль в жизни людей: они используются для автоматизации рутинных, помощи в трудных интеллектуальных задачах, обеспечения совместной работы и коммуникации. Информационные компьютерные технологии широко применяются в научной среде для сохранения и обработки информации с целью управления знаниями. Сюда также можно отнести каталогизацию и генерации новых знаний, обрательное их в научные труды, чтобы другие люди могли получить доступ и эффективно ими воспользоваться. Такое использование информационных технологий предоставляет широкие возможности и имеет большие перспективы, но, тем не менее, обладает рядом недостатков. На данный момент одна из ключевых проблем заключается в инструментарии, который служит для обеспечения сохранения и редактирования знаний. Он не является оптимальным по нескольким параметрам:

- простота переноса знаний из нецифровых источников в компьютерную систему;
- возможность переиспользования записанных знаний в статьях или иных научных трудах;
- автоматизация генерации, визуализации и прочей обработки знаний;
- представление знаний в хорошо воспринимаемом человеком формате;
- эффективность и интуитивная понятность пользовательского интерфейса;
- возможности для совместной работы и коллаборации.

Многие из этих проблем в той или иной степени решены в разных инструментах и компьютерных системах, однако на данный момент проблематично найти такую систему, которая решала бы все описанные выше задачи. В данной работе предлагается совместить один из современных подходов к редактированию текстовой информации (т.н. «блочное редактирование») с онтологическим подходом к структурированию знаний и

языком унифицированного смыслового представления знаний (SC-кодом) [1]. Так пользовательский интерфейс будет спроектирован посредством формализации предметной области компонентов пользовательского интерфейса, предметной области действий пользователя и предметной области внешних информационных конструкций и языков и разработкой решателя задач для пользовательского интерфейса [2]. Такой подход позволит реализовать концепцию адаптивного интерфейса, который будет подстраиваться под навыки и нужды пользователя, а также позволит без больших затрат времени интегрировать компоненты интерфейса, которые будут реализованы позже.

Был проведён сравнительный анализ редакторов исходных текстов баз знаний и редакторов текста в целом, и среди альтернатив были выделены следующие кандидаты: Overleaf [3], Notion [4], KBE [5], Protege [6]. По итогу анализа был выдвинут список требований к новому редактору баз знаний:

- Для обеспечения беспрепятственного ввода формализованных знаний необходим продуманный, эргономичный и задокументированный пользовательский интерфейс. Функции такого интерфейса включают в себя:
  - возможность использования горячих клавиш (в т.ч. возможности настроить персональные горячие клавиши);
  - графические методы редактирования текста (перетаскивание, контекстные меню, палетка команд и т.д.);
  - богатая обратная связь от компьютерной системы (реализация подсказок по наведению, документации и интерактивных инструкций по работе с программой);
  - активная помощь от компьютерной системы (реорганизация интерфейса в зависимости от текущих нужд пользователя или режима редактирования, проработанные сообщения об ошибках, механизм автодополнения текста).

– Для обеспечения возможности переиспользования записанных знаний в статьях или иных научных трудах необходим язык представления знаний, над которым определены операции перевода этого текста в требуемый формат и визуализацию. В случае описания онтологии компонентов пользовательского интерфейса редактора баз знаний на этом же языке представления знаний, появляется возможность реализовать реорганизацию интерфейса и механизмы помощи пользователю более оптимальным образом как с точки зрения программиста, так и с точки зрения пользователя. В качестве языка представления знаний предлагается использовать язык унифицированного представления знаний, SC-код. Это решение обеспечит следующие преимущества:

- Машина обработки знаний, при добавлении в неё соответствующих правил, сможет давать подсказки высокого качества и предлагать пользователю системы более разнообразные и персонализированные алгоритмы помощи в написании научных статей.
- Описание пользовательского интерфейса на языке унифицированного представления знаний позволит более просто менять техническую реализацию интерфейса и его свойств, т.к. для этого достаточно будет лишь описать внешний язык представления интерфейса и правила представления компонентов интерфейса на этом внешнем языке.
- Выбор SC-кода в качестве языка представления знаний даёт возможность интегрировать редактор с любыми *ostis*-системами, т.е. расширение функционала редактора станет доступной задачей даже для людей не знакомых с программированием. Сам же редактор можно будет использовать в составе более масштабных *ostis*-систем.

Помимо прочего, использование SC-кода позволит решить задачу автоматизации генерации, визуализации и другой обработки знаний, т.к. машина обработки знаний позволяет использовать поиск по графовым шаблонам, использовать подсистему логического вывода, в т.ч. по настраиваемым правилам, которые представлены на том же языке что и знания, над которыми проводятся операции. Таким образом операционная нагрузка

на научных работников будет уменьшена, т.к. необходимые операции обработки и вывода знаний нужно будет описать лишь единожды.

Представление знаний в хорошо воспринимаемом человеком виде и эффективность, понятность пользовательского интерфейса будут ключевыми целями при разработке пользовательского интерфейса. Реализации этих требований будет способствовать использование концепции блочного редактирования текста. Эта концепция стала широко применяться в редакторах текстов, которые были разработаны недавно, из-за того, что она предоставляет широкие и интуитивно-понятные возможности для реорганизации текста и выражения отношений между параграфами. Суть концепции заключается в том, что каждая обособленная единица информации выделяется в отдельный блок, над которым можно производить преобразования – поменять порядок следования блоков, сделать блок подблоком другого блока, вставить ссылку или отображение блока в другой блок. Для онтологического подхода к структурированию знаний наиболее важной является возможность выражать отношения между блоками с помощью концепции «подблок блока». В сочетании с SC-кодом, в котором заданы различные виды отношений между сущностями и имеется возможность описать пользовательские отношения для заданной предметной области, блочное редактирование должно стать мощным инструментом при редактировании научных статей или онтологий. Из всего вышесказанного следует, что разработка подобного редактора принесёт большой прирост в эффективности работы над трудными интеллектуальными задачами, а потому разработка целесообразна и перспективна.

#### СПИСОК ЛИТЕРАТУРЫ

1. Технология комплексной поддержки жизненного цикла семантически совместимых интеллектуальных компьютерных систем нового поколения / Под ред. В. В. Голенкова. – Минск: Бестпринт, 2023. – 1064 с.
2. Открытые семантические технологии проектирования интеллектуальных систем : сб. научн. тр. / Белор. гос. ун-т информ. и радиоэлектрон. ; редкол.: В. В. Голенков (отв. ред.) [и др.]. – Минск. : Бестпринт, 2021. – 330 с.
3. Overleaf documentation [Электронный ресурс]. – Режим доступа: <https://www.overleaf.com/learn>. – Дата доступа: 26.10.2023.
4. Notion. – Режим доступа: <https://www.notion.so/>. – Дата доступа: 26.10.2023.
5. KBE, Knowledge Base source Editor. – Режим доступа: <https://github.com/ostis-ai/kbe>. – Дата доступа: 26.10.2023.
6. Protege. – Режим доступа: <https://protege.stanford.edu/>. – Дата доступа: 26.10.2023.