



# OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

## КОМПОНЕНТНАЯ АРХИТЕКТУРА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ КОНСУЛЬТАЦИОННОГО ОБСЛУЖИВАНИЯ И ОБУЧЕНИЯ РАЗРАБОТЧИКОВ ПРОГРАММ

Пивоварчик О.В.\*

\* *Барановичский государственный университет, г. Барановичи, Республика Беларусь*

pivovarchyk@tut.by

В данной статье обоснована необходимость создания интеллектуальных компьютерных средств обучения в области программирования, имеющих открытую многокомпонентную архитектуру. В качестве компьютерного средства такого вида предложена интеллектуальная система консультационного обслуживания и обучения. В статье представлена общая компонентная архитектура системы и семантические модели каждого ее компонента.

**Ключевые слова:** интеллектуальная система; консультационная система; обучающая система; разработка программ; база знаний.

### Введение

Множество существующих методов и средств разработки программ, ориентированных на обработку баз знаний, а также постоянное их расширение и совершенствование, требуют от разработчиков не только глубоких знаний и умений, но и способностей быстро изучать новые технологии. Эффективными помощниками в изучении являются интеллектуальные компьютерные средства обучения (далее – КСО). В настоящее время в области технологий разработки программ существует множество интеллектуальных КСО. К числу наиболее упоминаемых в современных работах относятся интеллектуальные обучающие системы (далее – ИОС) и интеллектуальные учебные среды (далее - ИУС). ИОС основаны на интеграции технологий компьютерного обучения и экспертных систем и предназначены для освоения методов решения слабо структурированных задач [Попов и др., 1996]. ИУС предназначены для формирования навыков разработки программ.

Основными направлениями развития ИОС и ИУС являются использование технологий адаптации и многоагентных технологий, компонентно-ориентированная разработка, веб-ориентация. Примерами систем в области разработки программного обеспечения являются: ELM-ART, ACT-R Programming Tutor, GREATERP, LISP-TUTOR, FLEX, InterBook, QuizGuide (обеспечивают изучение функциональных языков);

VC Prolog Tutor (обеспечивает изучение языков логического программирования); J-LATTE, CITS, JITS, BITS (обеспечивают изучение объектно-ориентированных языков).

Актуальным является развитие специализированных инструментов, поддерживающих полный жизненный цикл создания и сопровождения ИОС и ИУС. Следует отметить, что в настоящее время разработано немного таких инструментов, примерами которых являются: МОНАП, АТ-ТЕХНОЛОГИЯ, KBS-HYPERBOOK.

Основным недостатком существующих ИОС и ИУС является архитектура. Большинство существующих систем создано как автономные программные продукты, которые не могут быть использованы в качестве компонентов других систем. Необходимо использовать либо целую систему, либо ничего. Небольшое число систем поддерживает компонентно-ориентированную архитектуру способную интегрироваться с другими системами. Однако, их интеграция возможна при условии использования одинаковых технологий и только при проектировании одной командой разработчиков, т.е. они не являются открытыми системами. Отсутствие возможности использования отдельно компонентов различных систем не позволяет создавать открытые распределенные архитектуры, усложняет процесс расширения функциональности.

Вторым недостатком является монофункциональность большинства отдельно

взятых систем. Например, QuizGuide позволяет проводить тестирование пользователя, InterBook предоставляет справочную информацию, ELM-ART проводит курс обучения языку программирования. Для обеспечения всестороннего индивидуального обслуживания пользователя, повышения удобства использования и эффективности решения задач необходимо объединение функциональности систем. Это возможно обеспечить путем создания одной многофункциональной системы, либо интеграцией систем или их компонентов.

Третьим недостатком является ограниченная функциональность при взаимодействии со средами программирования. Большинство существующих систем обеспечивает предоставление справочной информации по запросу и не имеет возможности проводить обучение и консультирование на основании информации о пользователе. Лишь единицы обеспечивают консультирование пользователя. Например, система FLEX [Стефанюк и др., 2004], [Стефанюк, 2007]. С другой стороны, лишь очень малое число ИУС включает компоненты среды программирования.

Решением данных проблем является разработка обучающих средств, имеющих многокомпонентную архитектуру. Технология разработки должна позволять повторно использовать компоненты, интегрировать их с другими компонентами, построенными с помощью как данной, так и других технологий. Также она должна быть открытой, чтобы обеспечить возможность использования компонентов различными командами разработчиков.

В данной статье представлена компонентная архитектура интеллектуальной системы консультационного обслуживания и обучения разработчиков программ, а также, общие принципы построения ее компонентов. Система проектируется на базе *Технологии OSTIS* (Open Semantic Technology for Intelligent Systems) [OSTIS], [IMS]. Формальные модели интеллектуальных систем, проектируемых на базе OSTIS, строятся на основе семантических сетей, что создает необходимые условия для обеспечения семантической совместимости интеллектуальных систем и их компонентов [Голенков и др., 2013].

## 1. Компонентная архитектура

Интеллектуальная система, спроектированная по *Технологии OSTIS*, представляет собой интеграцию *многократно используемых компонентов баз знаний, многократно используемых компонентов машин обработки знаний и многократно используемых компонентов интерфейсов*. Архитектура дочерней интеллектуальной системы включает:

- Многократно используемые компоненты, обеспечивающие базовые функции интеллектуальной системы – ядро базы знаний, ядро машины обработки знаний (набор базовых

sc-агентов, обеспечивающих редактирование базы знаний, навигацию по базе знаний, базовые информационно-поисковые операции), ядро интерфейсов.

- Многократно используемые компоненты, расширяющие базовую функциональность до требований, предъявляемых к интеллектуальной системе. Необходимые компоненты могут существовать в *Библиотеке многократно используемых компонентов OSTIS*, а также могут разрабатываться самостоятельно.

Описание базовых компонентов представлено в работах [Гракова и др., 2014], [Шункевич, 2014], [Корончик, 2014]. Задачей данной работы является описание компонентов, требуемых для функционирования интеллектуальной системы консультационного обслуживания и обучения разработчиков программ.

На основе анализа работ [Стефанюк, 2002], [Brusilovsky, 2012], [Рыбина, 2008], а также существующих систем были выделены основные структурные компоненты ИОС: эксперт по предметной области (или модель предметной области), модель обучаемого, педагог (или модель учебного материала), интерфейс. На основании обобщенной структурной модели и требуемой функциональности был определен набор необходимых компонентов для создания интеллектуальной системы консультационного обслуживания и обучения:

- *многократно используемые компоненты баз знаний*, описывающие предметную область;
- *многократно используемые компоненты баз знаний*, описывающие учебный материал;
- *многократно используемые компоненты баз знаний*, описывающие модель пользователя интеллектуальной системы;
- *многократно используемые компоненты баз знаний*, описывающие стратегии управления обучением и консультированием;
- *многократно используемые компоненты машин обработки знаний*, обеспечивающие обработку модели пользователя;
- *многократно используемые компоненты машин обработки знаний*, обеспечивающие управление обучением и консультированием;
- *многократно используемые компоненты машин обработки знаний*, обеспечивающие мониторинг и анализ деятельности пользователя;
- *многократно используемые компоненты интерфейсов*, обеспечивающие взаимодействие интеллектуальной системы с пользователем, преподавателем, экспертом;
- *многократно используемые компоненты интерфейсов*, обеспечивающие взаимодействие интеллектуальной системы с другими

интеллектуальными системами, другими платформами.

Каждый компонент может включаться в любую другую интеллектуальную систему.

## 2. Многократно используемые компоненты баз знаний

*Многократно используемые компоненты баз знаний*, описывающие предметную область, являются *неатомарными многократно используемыми компонентами*.

Предметная область может быть декомпозирована на подобласти, каждая из которых представляется в виде отдельного компонента. Например, предметная область разработка программного обеспечения включает множество используемых концептуальных, методических и технических средств. На основании анализа литературы была выделена следующая структура предметной области:

- теория программ
- инструменты:
  - языки программирования;
  - языки представления знаний;
  - среды разработки программ;
  - библиотеки типовых многократно используемых компонентов;
- методы и методики разработки программного обеспечения;
- управление разработкой программного обеспечения;
- оценка качества программного обеспечения;
- методы и методики обучения разработке программ.

Для каждой подобласти необходимо разработать несколько типов компонентов баз знаний. Для обучающих систем основными типами компонентов являются семантически совместимые онтологии предметных областей и/или онтологии задач, решаемых в предметной области. Были выделены следующие онтологии, которые необходимы для функционирования системы:

- *Терминологическая онтология* – спецификация терминов (понятий и отношений), используемых для описания предметной области.
- *Теоретико-множественная онтология* – онтология, содержащая теоретико-множественные связи (отношения) между понятиями и отношениями, определенными *терминологической онтологией*. К отношениям относятся: *включение\**, *разбиение\**, *объединение\**, *пересечение\**, *разность множеств\**, *область определения\**, *домен\**, *функция\**.

• *Информационная онтология* – спецификация структуры знаний, используемых для хранения информации о терминах. Для описания каждого ключевого понятия предметной области используются отношения: *определение\**, *пояснение\**, *детальное пояснение\**, *комментарий\**, *пример\**, *обучающая информация\**.

• *Статическая онтология* – описывает понятия, их атрибуты и отношения между ними. Используется для описания неизменяемых в некоторый период времени понятий.

• *Динамическая онтология* – используется для описания понятий, изменяемых во времени. Для описания понятий необходимо использовать процедурное представление знаний. Примерами общих терминов, включаемых в данную категорию, являются состояния и переходы из одного состояния в другое.

• *Эталонная модель предметной области* – онтология, используемая для моделирования уровня знаний и уровня умений пользователя (для обучения) или интересов пользователя (для консультирования и предоставления справочной информации).

Компонент эталонная модель предметной области формируется с целью использования обучающими и консультирующими системами. В эталонную модель включаются *sc-элементы*, которые отражают суть предметной области с точки зрения определенного объекта (обучающей системы, обучающего курса и др.). Она изоморфна целям (или цели) объекта.

Для систем, целью которых является обучение пользователя, онтология представляет собой максимальный уровень знаний и/или умений, который можно достигнуть. Знания и умения представляются в виде иерархии. При изучении предметной области модель знаний и умений пользователя структурируется в соответствии с данной иерархией. Предметная область считается изученной, если пользователь достиг самого высокого уровня иерархии. Знания входят во множество *sc-элементов знаний предметной области*. Умения входят во множество *sc-элементов умений предметной области*.

Для систем, целью которых является предоставление информации, онтология представляет собой множество возможных интересов в предметной области. Интересы также могут представляться в виде иерархии. Считается, что пользователю интересна предметная область, если уровень его интересов совпадает с самым высоким уровнем иерархии. Чем более сильный уровень интересов пользователя в предметной области, тем менее подробную информацию ему необходимо представлять о понятиях и тем больше понятий из данной предметной области ему необходимо выводить при запросе к разным

предметным областям. Интересы входят во множество *sc-элементов интересов предметной области*. Множества *sc-элементов знаний предметной области* и *sc-элементов интересов предметной области* могут пересекаться и быть равными.

*Многokrратно используемые компоненты баз знаний*, описывающие предметную область, разрабатываются экспертом и хранятся в Библиотеке *многokrратно используемых компонентов Технологии OSTIS*. При формировании обучающей системы некоторые компоненты уже могут существовать в библиотеке. В таком случае они просто интегрируются в интеллектуальную систему.

*Многokrратно используемые компоненты баз знаний*, описывающие учебный материал, могут быть *неатомарными многokrратно используемыми компонентами* или *атомарными многokrратно используемыми компонентами*. Компонент представляет собой *sc-структуру*, которая описывает некоторый учебный объект (обучающий курс, система тестирования, тренажер и др).

Рассмотрим формальную спецификацию компонента обучающий курс. Она включает следующие характеристики:

- Наименование курса на различных естественных языках. Знак компонента связан с *sc-ссылкой*, содержимым которой является наименование, отношением *наименование\**.
- Информация об авторе. Знак компонента связан со знаком автора отношением *автор\**. Автором может быть физическое лицо, коллектив, другая интеллектуальная система.
- Информация о цели или множестве целей компонента. Знак компонента связан с элементом множества *целей* отношением *цель обучающего курса\**.
- Множество ключевых понятий предметной области, которые доступны для изучения в рамках *обучающего курса*. Знак компонента связан с множеством *sc-элементов* отношением *ключевые sc-элементы обучающего курса\**.
- Множество других обучающих курсов, разделов, тестов и др., которые должны быть изучены, выполнены до начала изучения данного объекта. Связь компонента со знаком предварительного условия при помощи отношения *предварительное условие\**. Отношение является строгим, его выполнение обязательно.
- Информация о последовательности предоставления информации для изучения. Единицей информации является элемент множества *раздел обучающего курса*. Разделы могут составлять иерархию. Самый низкий уровень иерархии является минимальной единицей учебного материала. Разбиение разделов по уровням иерархии осуществляется

посредством отношения *подраздел\**. При выводе учебной информации по заданному разделу также включаются разделы более низких уровней иерархии. Последовательность изучения разделов определяется отношением *приоритет\**. Оглавление обучающего курса формируется на основании отношений *подраздел\** и *приоритет\**. Знак раздела может быть связан со знаками других разделов или обучающих курсов отношением *предварительное условие\**. Фрагмент разбиения обучающего курса на разделы представлен на рисунке 1.

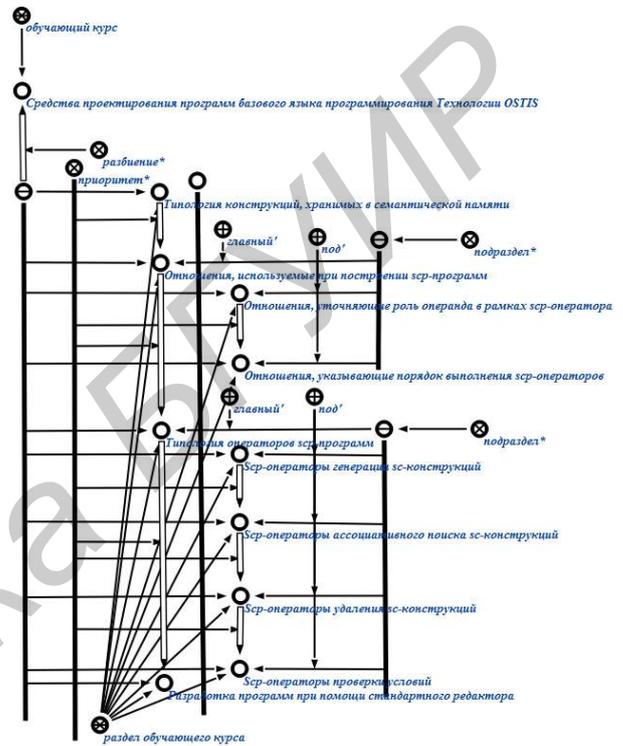


Рисунок 1 – Фрагмент разбиения обучающего курса на разделы

- Спецификация *раздела обучающего курса* включает:
  - Наименование раздела. Связь посредством отношения *наименование\**.
  - Множество ключевых понятий предметной области, которые доступны для изучения в рамках *раздела обучающего курса*. Связь посредством отношения *множество ключевых элементов раздела\**. Множество ключевых элементов раздела является подмножеством ключевых элементов обучающего курса. Учебный материал формируется из содержимого *sc-элементов*, связанных со знаком ключевого понятия отношениями: *определение\**, *пояснение\**, *детальное пояснение\**, *комментарий\**, *пример\**, *обучающая информация\**. Выбор связки для вывода учебной информации осуществляется на основании модели пользователя.

- Множество тестов для определения уровня усвоения раздела пользователем. Связь посредством отношения *тест раздела\**.

*Обучающий курс* может быть декомпозирован на множество более простых курсов (подкурсов). Для курсов, которые разбиваются на подкурсы, наличие информации о разделах не обязательно. Для курсов самого низкого уровня иерархии наличие информации о разделах обязательно. Курс считается изученным, если достигнуты его цели, а также цели всех подкурсов.

*Множественно используемые компоненты баз знаний*, описывающие модель пользователя интеллектуальной системы, являются *неатомарными* *множественно используемыми компонентами*. Для представления пользователя используется компонент, представленный в виде *sc-окрестности*, характеризующей его формальную спецификацию, а также используются компоненты в виде *статической онтологии* и *динамической онтологии*. Пользователем интеллектуальной системы может быть физическое лицо, коллектив, другая интеллектуальная система.

Рассмотрим *sc-окрестность* объекта физическое лицо. Она включает:

- Идентификационные данные.
- Общую информацию: персональные данные (фамилия, имя, отчество), возраст, пол и др.
- Информацию о психологических характеристиках.
- Информацию об индивидуальных особенностях, связанных с динамическими аспектами деятельности.
- Информацию об индивидуальных особенностях, связанных с процессами познания и мышления.
- Информацию о способностях.
- Лингвистические характеристики.
- Информацию о целях. Цели могут быть иницированными, т.е. подлежащими достижению в текущий период времени, и неиницированными. Иницированные цели отмечаются атрибутом *иницированный'*.
- Представление уровня знаний, умений или интересов. Для представления используется оверлейная модель, которая описывает знания, умения и интересы как подмножество *эталонной модели предметной области*. Уровень знаний, умений и интересов оценивается по заданной абсолютной или вероятностной шкале. Оценка уровня знаний может определяться в ходе изучения учебного материала, тестирования пользователя или решения задач. Оценка уровня умений определяется в ходе тестирования пользователя или в ходе решения задач. Оценка степени интереса определяется из количества запросов к данному *sc-элементу*.

- Информацию о совершенных ошибках.
- Историю взаимодействия пользователя с интеллектуальными системами.

Для представления формальной спецификации в *sc-коде* используется следующий набор *sc-элементов*: *полное имя\**, *имя'*, *фамилия'*, *отчество'*, *возраст\**, *пол*, *эмоциональный тип*, *тип темперамента*, *когнитивная способность*, *способность\**, *лингвистические характеристики\**, *родной язык'*, *знакомый язык'*, *предпочитаемый язык'*, *цель взаимодействия\**, *уровень знаний\**, *уровень умений\**, *интересы\**, *история пользователя\**.

Фрагмент формальной спецификации, характеризующей компонент физического лица, представлен на рисунке 2.

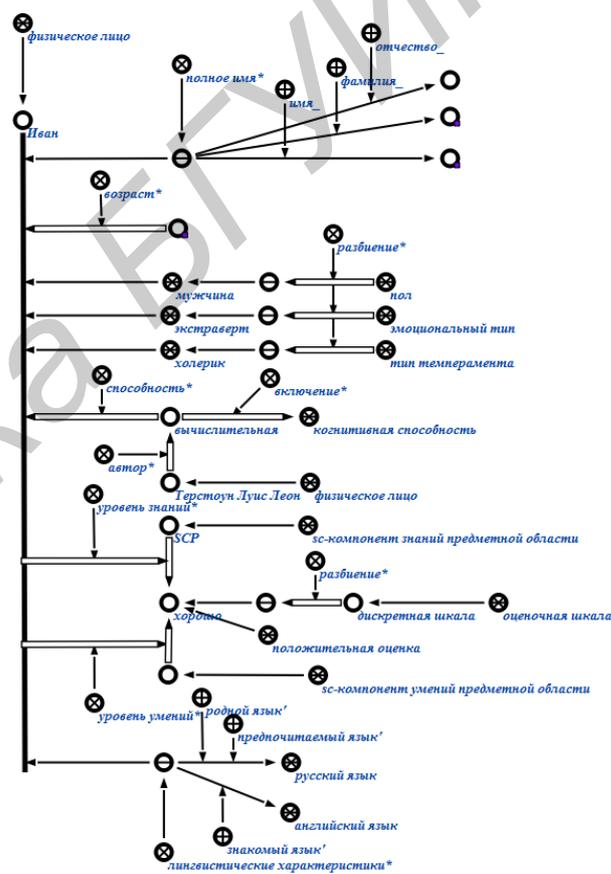


Рисунок 2 – Фрагмент формальной спецификации компонента

Одна часть из представленных характеристик не изменяется в различные периоды времени (статические характеристики), другая часть может изменяться (динамические характеристики). Изменяемые характеристики могут быть зависимы от процесса взаимодействия с интеллектуальной системой и не зависимы. Для описания данных характеристик используются компоненты: *статическая онтология* и *динамическая онтология*.

*Множественно используемые компоненты баз знаний*, описывающие стратегии управления обучением и консультированием, являются *неатомарными* *множественно используемыми компонентами* или *атомарными* *множественно*

используемыми компонентами. Компонент представляет собой шаблоны нескольких взаимосвязанных *sc-структур*, которые описывают последовательность обучения и применение обучающих воздействий к пользователю. *Sc-структуры* создаются в *sc-памяти* в процессе взаимодействия пользователя и интеллектуальной системы.

Центральным элементом *sc-структур* является пользователь, который помечен как активный. Активным может быть только зарегистрированный пользователь. Каждый зарегистрированный пользователь связан с ней отношением *пользователь\**. Включение знака пользователя во множество пользователей интеллектуальной системы подразумевает наличие знаний о нем. Знания могут быть различного рода в зависимости от вида объекта, которым является пользователь. Например, если пользователем является физическое лицо, то обязательно наличие персональной информации, индивидуальных характеристик, квалификации.

Первой формируемой *sc-структурой* является сеанс пользователя. Он включает знак интеллектуальной системы, знак активного пользователя, *sc-ссылки*, содержащие дату и время начала и окончания сеанса работы, знак последовательности действий пользователя в рамках сеанса. Роли *sc-ссылок* задаются атрибутивными отношениями *время начала'*, *время окончания'*. *Sc-структура* существует только в момент работы пользователя с системой, после завершения сеанса она становится элементом коротежа отношения *история пользователя\**.

Второй формируемой *sc-структурой* является сценарий обучения, который определяет стратегию или множество стратегий для обучения пользователя. Он связывает знак активного пользователя, знак обучающего курса, статический сценарий и/или динамический сценарий обучения под атрибутами *статический'*, *динамический'*. Статический сценарий обучения формируется в виде *sc-конструкции*, которая включает последовательность пар «обучающее воздействие – элементы раздела обучающего курса». Динамический сценарий – в виде *scr-программы*. *Scr-программа* формирует последовательность пар в процессе обучения.

Формальная спецификация компонента задается следующими множествами и отношениями: *пользователь\**, *активный пользователь\**, *сеанс пользователя\**, *время начала'*, *время окончания'*, *последовательность*, *начало'*, *окончание'*, *следующий'*, *доступный режим взаимодействия\**, *сценарий обучения\**, *обучающее воздействие*, *задача*, *действие*, *способы обратной связи\**, *ознакомление с предметной областью\**, *решение задач\**, *активные задачи\**, *шаблоны решений\**, *объяснение решения задачи\**, *тестирование\**.

### 3. Многократно используемые компоненты машин обработки знаний

*Многократно используемые компоненты машин обработки знаний*, обеспечивающие обработку модели пользователя, являются *платформенно-независимым многократно используемыми компонентами*, *неатомарными многократно используемыми компонентами*. Компоненты включают множество *абстрактных sc-агентов*, обеспечивающих формирование, модификацию, анализ модели пользователя, а также предоставление знаний о пользователе другим компонентам. В соответствии с функциональными требованиями были выделены следующие *sc-агенты*:

- *sc-агент формирования (первичного заполнения) модели пользователя;*
- *sc-агент обучения модели пользователя;*
- *sc-агент определения уровня знаний пользователя;*
- *sc-агент определения уровня умений пользователя;*
- *sc-агент определения интересов пользователя;*
- *sc-агент определения мотивации пользователя;*
- *sc-агент запроса уровня знаний пользователя;*
- *sc-агент запроса уровня умений пользователя;*
- *sc-агент запроса интересов пользователя.*

Первичное формирование модели пользователя производится при регистрации. Затем изменение модели производится интеллектуальной системой. По запросу корректировку модели может выполнять преподаватель или сам пользователь.

*Многократно используемые компоненты машин обработки знаний*, обеспечивающие управление обучением и консультированием, являются *платформенно-независимым многократно используемыми компонентами*, *неатомарными многократно используемыми компонентами*. Компоненты включают множество *абстрактных sc-агентов*, обеспечивающих управление обучением и консультированием. Основной функцией компонентов является управление прохождением сценария обучения и формирование сеанса пользователя. В соответствии с функциональными требованиями были выделены следующие *sc-агенты*:

- *sc-агент запуска сеанса пользователя;*
- *sc-агент выбора режима взаимодействия;*
- *sc-агент формирования сценария обучения;*
- *sc-агент прохождения обучающего курса;*

- *sc-агент ознакомления с ключевыми понятиями;*
- *sc-агент тестирования;*
- *sc-агент решения задачи;*
- *sc-агент применения обучающих воздействий;*
- *sc-агент объяснения решения задачи;*
- *sc-агент контроля правильности разработанных программ;*
- *sc-агент вывода комментариев;*
- *sc-агент локализации ошибок;*
- *sc-агент рекомендации обучающих курсов;*
- *sc-агент определения уровня сложности обучающего курса;*
- *sc-агент корректировки сценария обучения;*
- *sc-агент завершения сеанса пользователя.*

Множественно используемые компоненты машин обработки знаний, обеспечивающие мониторинг и анализ деятельности пользователя, являются платформенно-независимыми множественно используемыми компонентами, неатомарными множественно используемыми компонентами. Компоненты включают множество абстрактных *sc-агентов*, осуществляющих мониторинг и анализ действий пользователя.

Мониторинг и анализ осуществляются на предмет соответствия действия или множества действий предыдущим ошибкам данного пользователя, типичным ошибкам пользователей, а также на предмет возникновения некоторых критических ситуаций. Множество типичных ошибок и шаблоны критических ситуаций оформляются в виде отдельных компонентов баз знаний. В случае возникновения таких ситуаций корректировка действий пользователя осуществляется компонентом, обеспечивающим управление обучением и консультированием. Наличие их расширяет функциональность системы, повышает комфорт работы пользователя, однако, не обязательно. В соответствии с функциональными требованиями были выделены следующие *sc-агенты*:

- *sc-агент сбора данных о действиях пользователя;*
- *sc-агент запроса ошибок пользователя;*
- *sc-агент запроса типичных ошибок;*
- *sc-агент запроса критических ситуаций;*
- *sc-агент оценки действий пользователя.*

#### 4. Множественно используемые компоненты интерфейсов

Множественно используемые компоненты интерфейсов, обеспечивающие взаимодействие интеллектуальной системы с пользователем,

преподавателем, экспертом или с другими интеллектуальными системами, другими платформами, могут быть платформенно-независимым множественно используемыми компонентами или платформенно-зависимым множественно используемыми компонентами.

Такие компоненты обладают своей спецификой и их реализация зависит от требований конкретной интеллектуальной системы. Некоторые из требуемых компонентов могут существовать в Библиотеке множественно используемых компонентов Технологии OSTIS. В таком случае они просто интегрируются в интеллектуальную систему.

#### Заключение

В данной статье обоснована необходимость создания компонентно-ориентированных открытых ИОС и ИУС в области разработки программ, ориентированных на обработку баз знаний, что позволит расширить круг разработчиков. А также, показана необходимость их интеграции со средами разработки программ, что обеспечит не только дополнительное удобство разработчику, но и поспособствует росту его квалификации.

В качестве КСО, удовлетворяющего вышеперечисленным требованиям, предлагается интеллектуальная система консультационного обслуживания и обучения, построенная на базе Технологии OSTIS. В работе представлена компонентная архитектура системы. А также, семантические модели множественно используемых компонентов баз знаний, машин обработки знаний и интерфейсов.

#### Библиографический список

- [Попов и др., 1996] Попов Э.В., И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. Статические и динамические экспертные системы, учебное пособие. - М.: «Финансы и статистика», 1996. -318 с.
- [Стефанюк, 2004] Стефанюк, В. Л. Локальная организация интеллектуальных систем / В. Л. Стефанюк. - М.: ФИЗМАТЛИТ, 2004. - 328 с.
- [Стефанюк и др., 2007] Стефанюк, В. Л. Сотрудничающий компьютер : монография / В. Л. Стефанюк, А. В. Жожикашвили. - М.: Наука, 2007. - 274 с.
- [OSTIS] Открытая семантическая технология компонентного проектирования интеллектуальных систем [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://ostis.net/mediawiki/index.php/>.
- [IMS] Открытая семантическая технология компонентного проектирования интеллектуальных систем [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://ims.ostis.net/>.
- [Голенков и др., 2013] Голенков, В. В. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск : БГУИР, 2013. - С. 55-78.
- [Гракова и др., 2014] Гракова, Н. В. База знаний интеллектуальной метасистемы поддержки Проектирования интеллектуальных систем / Н. В. Гракова, И. Т. Давыденко, К. В. Русецкий // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». - Минск: БГУИР, 2014. - с. 83 - 92.

[Шункевич, 2014] Шункевич, Д. В. Машина обработки знаний интеллектуальной метасистемы поддержки проектирования интеллектуальных систем / Д. В. Шункевич // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2014. – с. 93 – 96

[Корончик, 2014] Корончик, Д. Н. Пользовательский интерфейс интеллектуальной метасистемы поддержки проектирования интеллектуальных систем / Д. Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2014. – с. 79 – 82.

[Стефанюк, 2002] Стефанюк, В. Л. Учить или учиться? / В. Л. Стефанюк // Новости искусственного интеллекта. – 2002. – №5. – с. 13-24.

[Brusilovsky, 2012] Brusilovsky, P. Adaptive Hypermedia for Education and Training / P. Brusilovsky // Adaptive Technologies for Training and Education. – Cambridge : Cambridge University Press, 2012. – p. 46-68.

[Рыбина, 2008] Рыбина, Г.В. Обучающие интегрированные экспертные системы: некоторые итоги и перспективы / Г.В. Рыбина // Искусственный интеллект и принятие решений. – 2008. – №1. – с. 22-46.

## A COMPONENT-BASED ARCHITECTURE OF INTELLIGENT SYSTEMS OF CONSULTATION PROVIDING AND TRAINING OF SOFTWARE DEVELOPERS

Pivovarchyk O.

\* *Baranovich State University, Baranovich,*  
*Belarus*  
pivovarchyk@tut.by

The given article explains the necessity of creation of intelligent computer means of training, having open multicomponent architecture. An intelligent system of consultation providing and training is offered as a computer means of this kind. The article presents general component architecture of intelligent system and semantic models of each component.

### Introduction

Continuous progress of methods and tools for software development requires deep knowledge, skills and ability from programmers and quickly learn new technologies. Computer training tools are effective helpers in learning new technologies. Currently, there are many computer training tools in developing software for the processing of knowledge bases. The main disadvantages of the existing systems are architecture, monofunctionality, limited functionality when interacting with development environments. Solving problems is the development of learning tools with multi-component architecture.

The article presents a component-based architecture of intelligent systems of consultation providing and training of software developers. The system is designed on the basis of Technology OSTIS.

### Main Part

An intelligent system designed on the basis of Technology OSTIS consists of reusable components of knowledge bases, reusable components of knowledge processing machines and reusable components of

interfaces. The architecture of the child's intellectual system includes:

- Reusable components that provide the basic functions of an intelligent system (the core of the knowledge base, the core of the knowledge processing machine, the core of interfaces).
- Reusable components that extend the basic functionality of an intelligent system. The necessary components can exist in *The Library of OSTIS reusable components* or can be developed.

Components that extend the basic functionality of an intelligent system of consultation providing and training of software developers are:

- reusable components of knowledge bases that describe the knowledge domain;
- reusable components of knowledge bases that describe an educational material;
- reusable components of knowledge bases that describe the user model of an intelligent system;
- reusable components of knowledge bases that describe strategies of training and consulting management;
- reusable components of knowledge processing machines that provide processing of an user model;
- reusable components of knowledge processing machines that provide training and consulting management;
- reusable components of knowledge processing machines that provide monitoring and analysis of user activity;
- reusable components of interfaces for communication an intelligent system with users, experts, teachers;
- reusable components of interfaces for communication an intelligent system with the others systems.

Each component may be included in any other intelligent system.

### Conclusion

This article describes the necessity of the creation of component-oriented open Intelligent Training Systems and Intelligent Educational Environments that will expand the range of developers. It shows the necessity of their integration with development environments that will not only provide additional convenience to the programmers, but also will contribute to professional growth.

The component-based architecture of intelligent systems of consultation providing and training are proposed in this article. The semantic models of reusable components of knowledge bases, knowledge processing machines and interfaces are presented.