Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

ОТЧЁТ

по ознакомительной практике

Выполнил: Д. С. Семёнов

Студент группы
321703
Проверил: В. Н. Тищенко

Минск 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель:

Закрепить практические навыки формализации информации в интеллектуальных системах с использованием семантических сетей.

Задачи:

- Построение формализованных фрагментов теории интеллектуальных компьтерных систем и технологий их разработки;
- Построение формальной семантической спецификации библиографических источников, соответствующих указанным выше фрагментам;
- Оформление конкретных предложений по развитию текущей версии Стандарта интеллектуальных компьтерных систем и технологий их разработки

1 ПОСТАНОВКА ЗАДАЧИ

Часть 2 Учебной дисциплины "Представление и обработка информации в интеллектуальных системах"

⇒ *библиографическая ссылка**:

- *монография OSTIS*
- *Материалы конференций OSTIS*
- *М.П. Малыхина..МУЛЬТИАГЕНТНЫЕ СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА*
  ⇒ *URL**:
  [https://ntk.kubstu.ru/data/mc/0051/2074.pdf]
- *Д.А. Герасимов..МУЛЬТИАГЕНТНЫЕ СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА*
  ⇒ *URL**:
  [https://ntk.kubstu.ru/data/mc/0051/2074.pdf]
- *Определение и проблемы языков программирования*
  ⇒ *URL**:
  [https://ppt-online.org/149158]
- *Гардейчик С.М..МЕТАПРОГРАММИРОВАНИЕ КАК ОДИН ИЗ ВИДОВ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ*
  ⇒ *URL**:
  [https://rep.bntu.by/bitstream/handle/data/106728/7-8.pdf?sequence=1]
- *Корниевич А.И..МЕТАПРОГРАММИРОВАНИЕ КАК ОДИН ИЗ ВИДОВ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ*
  ⇒ *URL**:
  [https://rep.bntu.by/bitstream/handle/data/106728/7-8.pdf?sequence=1]
- *Кроссплатформенность*
  ⇒ *URL**:
  [https://secretmag.ru/enciklopediya/chto-takoe-krossplatformennost-obyasnyaem-prostymi-slovami.htm]
- *Шарикова М.И..Определение и проблемы языков программирования*
  ⇒ *URL**:
  [https://novainfo.ru/article/8173]
- *Ясницкий Л.Н..Интеллектуальные системы*
  ⇒ *URL**:
  [https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf]
- *Степанов О.Г..Автоматное программирование*
  ⇒ *URL**:
  [https://cyberleninka.ru/article/n/avtomatnoe-programmirovanie-s-ispolzovaniem-dinamicheskih-yazykov-programmirovaniya]

2 ФОРМАЛИЗОВАННЫЕ ФРАГМЕНТЫ ТЕОРИИ ИНТЕЛЛЕКТУАЛЬНЫХ КОМПЬЮТЕРНЫХ СИСТЕМ И ТЕХНОЛОГИЙ ИХ РАЗРАБОТКИ

мультиагентная система

:= [направление искусственного интеллекта, которое использует систему агентов для решения сверхсложных задач или глобальных проблем]

$\subset$ *искуственный интеллект*

$\Rightarrow$ *примечание**:

[агенты в мультиагентных системах взаимосвязаны между собой и обладают возможностью обмена сообщениями. У каждого агента есть своя цель и решаемая им задача. Обычно в мультиагентных системах используются программные агенты. Тем не менее, составляющими мультиагентных систем могут быть также роботы, люди или команды людей]

$\Rightarrow$ *обобщенная декомпозиция**:

основные требования при формировании мультиагентных систем

$=$ { • *автономность*

$\Rightarrow$ *пояснение**:

[Все агенты должны быть независимы в управлении, а поведение агентов прикладных систем определяется их ролью в этой системе. При этом агенты могут иметь свои индивидуальные цели, которые отличаются от целей системы. В соответствии с этим они способны самостоятельно планировать собственное поведение для достижения своих целей, осуществляя самоконтроль над своими действиями и внутренним состоянием, а также контролировать узлы связи агентов.]

• *децентрализация*

$\Rightarrow$ *пояснение**:

[Агенты наделяются правами и полномочиями для реализации автономного поведения и способности принимать решения. Таким образом, агенты могут выполнять не только команды извне, но и обмениваться сообщениями, на основе которых принимают решения. Иногда в виде исключения применяются посредники в виде агентов, ответственных за реализацию протокола передачи сообщений между агентами.]

• *взаимодействия*

$\Rightarrow$ *пояснение**:

[Автономность и децентрализация не исключают, а наоборот, предполагают активные взаимодействия между агентами. Поведение агентов предполагает обмен информацией, координацию действий, согласование решений и т.п. Способность к взаимодействию агентов имеет принципиальное значение, поскольку именно от нее в большинстве случаев зависит возможность обеспечения и качество реализации всех других свойств. Это так называемые "вычисления, основанные на взаимодействиях". Таким образом, достигается максимальная фокусировка агента на своей задаче и сокращается время работы агентной системы]

• *распределенность*

⇒ *пояснение**:

[Использование мультиагентных систем является естественным подходом для разработки приложений распределенных систем, при котором происходит разброс агентов на разных вычислительных устройствах или процессорах, что может существенно повышать производительность прикладных систем. Тем не менее, это вовсе не исключает преимуществ использования мультиагентного подхода для решения прикладных задач, когда агенты функционируют на одном вычислительном устройстве в параллельном режиме]

- *симметричность*

⇒ *пояснение**:

[При обмене сообщениями между агентами, каждому из них должны быть предоставлены одинаковые возможности участия в переговорах для организации корректного получения информации]

- *устойчивость*

⇒ *пояснение**:

[При обмене сообщениями между агентами каждый из них не должен позволять другим агентам извлекать дополнительную пользу при отклонении от установленных правил. Таким образом, обмен сообщениями должен обеспечить достижение состояния равноправия между агентами]

- *простота*

⇒ *пояснение**:

[Для повышения скорости обмена сообщениями и ускорения работы всей мультиагентной системы обмен сообщениями не должен предъявлять дополнительных требований к архитектуре агентов и усложнять структуру коммуникационных протоколов]

- *безопасность*

⇒ *пояснение**:

[При разработке мультиагентной системы должна обеспечиваться ее защищенность во избежание несанкционированного доступа к компьютерной информации. Несанкционированный доступ может повлечь за собой как сбой в работе мультиагентной системы, так и утрату или искажение важной информации]

- *интеллектуальность*

⇒ *пояснение**:

[Мультиагентная система должна отвечать требованиям интеллектуальности, то есть содержать в себе интеллектуальные методы. Для повышения эффективности можно применять базы данных с использованием интеллектуальных методов]

}

⇒ *автор**:

- *М.П. Малыхина*
- *Д.А. Герасимов*

⇒ *библиографическая ссылка**:

- *МУЛЬТИАГЕНТНЫЕ СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА*
- *https://ntk.kubstu.ru/data/mc/0051/2074.pdf*

язык программирования

:= [знаковая система для планирования поведения компьютера]

:= [правила образования знаков и согласованные с ними правила образования денотатов]

⇒ *примечание**:

[денотат - значение, смысл]

⇒ *примечание**:

[знак - обозначение или имя]

:= [средство общения между человеком и компьютером]

:= [инструмент для производства программных услуг]

⇒ *обобщенная декомпозиция**:

основные предметные области знаний применения языков программирования

= {
- *научные вычисления*
- *обработка деловой информации*
- *искусственный интеллект*
- *системная область*
- *Web-обработка*

}

⇒ *обобщенная декомпозиция**:

основные критерии эффективности языка программирования

= {
- *читабельность*
  - ⇒ *примечание**:

    [легкость понимания текста программ]
- *легкость создания программ*
  - ⇒ *примечание**:

    [удобство языка для создания программ в выбранной предметной области]
- *надежность*
  - ⇒ *примечание**:

    [минимум ошибок при работе с программой]
- *стоимость*
  - ⇒ *примечание**:

    [всего жизненного цикла программ - выполнения, трансляции, создания и тестирования, сопровождения]
- *переносимость программ*
  - ⇒ *примечание**:

    [всего жизненного цикла программ - выполнения, трансляции, создания и тестирования, сопровождения]
- *универсальность*
  - ⇒ *примечание**:

    [применимость к широкому кругу задач]
- *четкость*
  - ⇒ *примечание**:

    [полнота и точность официального описания языка]

}

⇒ *обобщенная декомпозиция**:

актуальные проблемы языков программирования

= {
- *аппаратная сложность*
  - ⇒ *примечание**:

[Аппаратная сложность опережает наше умение строить ПО, использующее потенциальные возможности аппаратуры.]

- *отставание разработки*

⇒ *примечание**:

[Умение разрабатывать новые программы отстает от требований к новым программам.]

- *качество существующих программ*

⇒ *примечание**:

[Возможности эксплуатировать существующие программы угрожает низкое качество их разработки.]

}

⇒ *библиографическая ссылка**:

- *Определение и проблемы языков программирования*
- *https://ppt-online.org/149158*

парадигма программирования

:= [исходная концептуальная схема или модель, принятая в качестве образца постановки проблем и их решения]

:= [результат анализа и обобщения способа прогрессивного научного мышления, характерного для данной области]

⇒ *примечание**:

[Впервые понятие парадигмы ввел в естествознание Т. Кун. Выбор той или иной парадигмы для решения конкретной проблемы определяет методологическое обоснование методов ее решения и задает регулятивные основы мышления разработчика.]

⇒ *разбиение**:

{
- *операционная*
- *императивная*
- *функциональная*
- *структурная*
- *объектно-ориентированная*
- *последовательная*
- *параллельная*
- *потоковая*
- *трансформационная*
- *асинхронная*
- *синхронная*
- *дефиниционная*
- *логическая*
- *демонстрационная*
- *сетевая*
- *локальная*
- *капсулизации*

}

⇒ *автор**:

- *В С. Выхованец*
- *О.Д. Выхованец*

⇒ *библиографическая ссылка**:
http://valery.vykhovanets.ru/Texts/1997/Vykhovanets1997_2.pdf

метапрограммирование

:= [вид программирования, связанный с созданием программ, порождающих другие программы как результат своей работы]

⇒ *примечание**:

[Метапрограммирование применяется для повышения функциональности программ ценой меньших затрат (объем кода, усилия на сопровождение и т.п.). Его характерной особенностью является то, что часть необходимых пользователю вычислений выполняется на этапе трансляции программы.]

⇒ *автор**:

- *Гардейчик С.М.*
- *Корниевич А.И.*

⇒ *библиографическая ссылка**:

- *МЕТАПРОГРАММИРОВАНИЕ КАК ОДИН ИЗ ВИДОВ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ*
- *https://rep.bntu.by/bitstream/handle/data/106728/7-8.pdf?sequence=1*

кроссплатформенность

:= [способность IT-продукта полноценно работать на любом устройстве вне зависимости от типа операционной системы]

⇒ *примечание**:

[Задача программистов, обеспечивающих кроссплатформенность — написать такой код, чтобы пользователи смогли получить доступ к сервису с устройств, работающих на любой операционной системе]

⇒ *библиографическая ссылка**:

https://secretmag.ru/enciklopediya/chto-takoe-krossplatformennost-obyasnyaem-prostymi-slovami.htm

семантический разрыв

:= [различие между описаниями объектов различных формальных представлений, например, языкового и символьного описаний]

:= [несоответствие выводимого результата запросу пользователя]

⇒ *примечание**:

[пользователь при формировании запроса оперирует понятиями высокого уровня, такими как имя объекта. При этом существующие системы поиска данных оперируют низкоуровневыми признаками, такими как цвет, форма и текстура]

⇒ *автор**:

Шарикова М.И.

⇒ *библиографическая ссылка**:

https://novainfo.ru/article/8173

экспертная система

:= [сложный программный комплекс, аккумулирующий знания специалистов в конкретной предметной области и использующий эти знания с целью выработки логически обоснованных рекомендаций и решений проблем, а также для консультаций менее квалифицированного пользователя]

⇒ *автор**:

Ясницкий Л.Н.

⇒ *библиографическая ссылка**:

- *Интеллектуальные системы*
- *https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf*

интерфейс разработчика

:= [программа, с помощью которой инженер-когнитолог и программист могут создавать базу знаний в диалоговом режиме. Включает в себя системы вложенных меню, шаблонов языка представления знаний, подсказок (help-режим) и других сервисных средств, облегчающих работу с базой знаний.]

⇒ *автор**:

Ясницкий Л.Н.

⇒ *библиографическая ссылка**:

- *Интеллектуальные системы*
- *https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf*

интерфейс пользователя

:= [комплекс программ, реализующих диалог пользователя с экспертной системой на стадии как ввода информации, так и получения результатов]

⇒ *автор**:

Ясницкий Л.Н.

⇒ *библиографическая ссылка**:

- *Интеллектуальные системы*
- *https://publications.hse.ru/pubs/share/folder/uxv237cikj/202053393.pdf*

SWITCH-технология

:= [автоматное программирование]

:= [программирование с явным выделением состояний]

⇒ *примечание**:

[Одной из основных SWITCH-технологий является графический язык, позволяющий описывать поведение систем со сложным поведением в терминах состояний и переходов между ними и связи между этими системами. При использовании SWITCH-технологии в разработке программного обеспечения важной частью разработки является реализация поведения, описанного для диаграммах переходов, на целевом языке программирования.]

⇒ *обобщенная декомпозиция**:

подходы для реализации поведения систем

= { • *полностью ручное программирование*

⇒ *примечание**:

[Одним из простейших общепринятых методов ручного программирования является следующий: текущее состояние системы хранится в переменной интегрального или перечислимого типа, а основная логика программы сосредоточена внутри одного или нескольких операторов *switch*, определяющих действия программы в зависимости от текущего состояния. Другим популярным методом является использование паттерна программирования State. Несмотря на такие преимущества этой группы методов, как высокая производительность и полный контроль над получаемым кодом, она имеет существенные недостатки — низкая читаемость кода и большая трудоемкость.]

• *автоматическая генерация кода по диаграмме переходов*

⇒ *примечание**:

[Обычно при этом подходе генерируется код, аналогичный тому, который получается при использовании ручного программирования. Недостатками этого подхода являются низкая читаемость кода (свя-

занная с тем, что в качестве целевого языка используется императивный язык, например *Java*, и при переносе теряется информация, специфичная для логики диаграмм переходов), малая степень контроля над получаемым кодом, невозможность ручного изменения получаемого кода и привязанность к конкретному формату входных данных, с использованием которого задается исходная диаграмма переходов]

- *Ручное написание кода с использованием специальной библиотеки*

⇒ *примечание**:

[Ручное написание кода с использованием специальной библиотеки. В этом случае происходит перенос диаграммы переходов в вызовы специальной библиотеки, которая по этим инструкциям строит внутреннее представление диаграммы переходов. Затем, согласно этому представлению, происходит исполнение логики описанного автомата. Основным преимуществом подхода является то, что вызовы библиотеки отражают семантику диаграммы переходов каждый вызов может, например, соответствовать объявлению состояния или перехода), что позволяет создавать читаемый код, который легко поддерживать. Также некоторые библиотеки ориентированы на конкретные виды взаимодействия автоматного и объектно-ориентированного кода, что позволяет более гладко комбинировать эти подходы к программированию. Основными недостатками являются низкая производительность некоторых из таких систем (существуют системы, основанные на метапрограммировании и статической генерации кода, которые позволяют достичь высокой производительности), а также невозможность описать ряд конструкций диаграмм переходов, используя ограниченный синтаксис целевого языка]

}

⇒ *автор**:
Степанов О.Г.

⇒ *библиографический источник**:
https://cyberleninka.ru/article/n/avtomatnoe-programmirovanie-s-ispolzovaniem-dinamicheskih-yazykov-programmirovaniya

ЗАКЛЮЧЕНИЕ

В рамках учебной практики были изучены и повышены навыки формализации научных текстов. Были формализованы понятия, связанные с изучаемой дисциплиной и темой "Проблемы текущего состояния в области разработки и применения языков программирования при помощи пакета макросов "scn-latex".

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Кормен, Д. Алгоритмы. Построение и анализ / Д. Кормен. — Вильямс, 2015. — С. 1328.

[2] Кузнецов, О. П. Дискретная математика для инженера / О. П. Кузнецов, Г. М. Адельсон-Вельский. — Энергоатомиздат, 1988. — С. 480.

[3] Оре, О. Теория графов / О. Оре. — Наука, 1980. — С. 336.

[4] Харарри, Ф. Теория графов / Ф. Харарри. — Эдиториал УРСС, 2018. — С. 304.

[5] Wooldridge, M. An introduction to multiagent systems / M. Wooldridge. — 2nd ed. — Chichester : J. Wiley, 2009. — 484 p.