

The background features a dark blue gradient with faint, light blue technical diagrams. On the left, there is a large circular scale with numerical markings from 150 to 260. Several concentric circles and dashed lines with arrows are scattered across the page, suggesting a technical or engineering theme.

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ГРАФИЧЕСКОГО АНАЛИЗА СООТВЕТСТВИЯ ОНТОЛОГИЙ ТРЕБОВАНИЯМ

Студент: Василейко А.В.

Руководитель: доц., к.ф.-м.н. Ермолаев В.А.

Цель

Разработка алгоритмического и программного обеспечения для конвертации представлений требований в описательные модели представлений знаний

- конвертировать на языке UML (диаграммы классов) в язык представлений знаний OWL
- сравнить формализованные модели предметных областей с требованиями в графическом виде используя среду Protégé
- выполнить анализ подходов к формализации требований к предметной области

Технологии, вклад

- Java (JDK) – язык программирования и среда выполнения
- UML / OWL / XMI
- ArgoUML – UML редактор с открытым исходным кодом
- Protégé (OWLDiff) – графическая среда для работы с OWL
- Apache Jena – библиотека для работы с OWL
- Использование Jena для конвертации
- Расширение ArgoUML (Exports to OWL...)

ArgoUML

The screenshot displays the ArgoUML application window. On the left, a menu is open, listing various actions such as 'New', 'New Profile', 'Open Project...', 'Save Project', 'Save Project As...', 'Revert to Saved', 'Import XML...', 'Export XML...', 'Export OWL...', 'Import Sources...', 'Page Setup...', 'Print...', 'Export Graphics...', 'Export All Graphics...', 'Notation', and 'Project Properties'. Below these, a list of files is shown: 'outputparser.xml', 'test_xmi.xml', 'main_rules.xml', and 'main_rules.owl'. The main workspace on the right contains a class diagram titled 'main_rules.xmi.zargo - Class Diagram - ArgoUML *'. The diagram illustrates relationships between several classes: 'TimeLine' (1) has 'TimeUnit' (1..*); 'TimeUnit' (1..*) is 'measuredUsing' by 'TemporalMeasure' (1); 'TemporalMeasure' (0..*) 'measures' 'Present' (1..*); 'Present' (1..*) is 'measuredBy' 'TimeInstant' (1); 'TemporalMeasure' (1) 'generates' 'Clock' (1..*); 'Clock' (1..*) 'uses' 'TemporalInstrument' (0..*); 'TemporalInstrument' (0..*) is 'associatedWith' 'Clock' (0..*); 'Clock' (0..*) is 'alignedWith' 'PhysicalClock' (0..*); 'PhysicalClock' (0..*) is a specialization of 'Clock' (0..*); 'LogicalClock' (0..*) is a specialization of 'Clock' (0..*); 'ClockRelation' (1) is a specialization of 'AffineClockRelation' (1); 'AffineClockRelation' (1) has attributes 'scaleFactor : float' and 'shift : float'; 'ClockRelation' (1) has a method 'newOperation()'. The bottom status bar indicates '3M used of 455M max'.

Основные результаты

- Разработаны эвристики и алгоритмы конвертации моделей знаний на языке UML в язык представления знаний OWL
- Программный продукт который реализует конвертацию и позволяет анализировать и редактировать UML диаграммы
- Важная часть системы по формализации знаний полученных из семантически насыщенных текстов



Преимущества, недостатки

Преимущества:

- анализ, редактирование, конвертация в рамках одной программы
- быстрая конвертация относительно больших UML диаграмм
- **практическая ценность, очень мало подобных решений**

Недостатки:

- в ArgoUML нет подсистемы плагинов и расширений
- не уведомляет о возможной потере данных при конвертации связей
- анализ и редактирование OWL происходит в отдельной среде Protégé

Вывод

- Разработано необходимое программное обеспечение
- Проведены эксперименты для проверки работоспособности ПО и корректности результатов конвертации UML в OWL
- Задание выполнено в полном объеме, цель достигнута