

article/n/biometriya-v-kontekste-personalnyh-dannyh-i-geneticheskoy-informatsii-pravovye-problemy/viewer.

2. Facial recognition technology (FRT) [Электронный ресурс]: [https://www.tadviser.ru/index.php/Статья:системы_распознавания_лиц_\(Facial_recognition\)](https://www.tadviser.ru/index.php/Статья:системы_распознавания_лиц_(Facial_recognition)).

ИСПОЛЬЗОВАНИЕ ДИАГРАММ СОСТОЯНИЙ UML НА ПРИМЕРЕ ПРОЕКТИРОВАНИЯ ЛЕКСИЧЕСКОГО АНАЛИЗАТОРА

А.Л. Корякин

Научный руководитель – Алексеев В.Ф., канд.техн.наук, доцент
Белорусский государственный университет информатики и радиоэлектроники

Диаграмма состояний UML представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий. Диаграмма состояний по существу является графом специального вида, который служит для предоставления некоторого конечного автомата. Хотя понятие конечного автомата основывается на общей теории автоматов, в контексте языка UML оно обладает дополнительной семантикой. Вершинами графа конечного автомата в контексте языка UML, являются состояния и некоторые другие типы элементов модели (псевдосостояния), которые изображаются соответствующими графическими символами. Дуги графа служат для обозначения переходов из состояния в состояние. Диаграммы состояний могут быть вложены друг в друга, образуя вложенные диаграммы для более детального представления отдельных элементов модели. Для понимания семантики конкретной диаграммы состояний необходимо представлять не только особенности поведения моделируемой сущности, но и знать общие сведения из теории конечных автоматов [1].

Конечный автомат (*state machine*) представляет собой некоторый формализм для моделирования поведения отдельных элементов модели или системы в целом. Конечный автомат описывает поведение отдельного объекта в форме последовательности состояний, которые охватывают все этапы его жизненного цикла, начиная от создания объекта и заканчивая его уничтожением. Формализм конечного автомата основан на выполнении следующих обязательных условий [1]:

1. Конечный автомат (КА) не запоминает историю перемещения из состояния в состояние.
2. В каждый момент времени конечный автомат может находиться только в одном из своих состояний
3. Длительность нахождения КА в том или ином состоянии никак не специфицируется.
4. Количество состояний КА конечно.
5. Для каждого состояния, кроме начального, должно быть определено хотя бы одно предшествующее состояние.
6. При наступлении одного и того же события моделируемый элемент не может перейти в два или более последующих состояния.

Работа лексического анализатора также задаётся некоторым конечным автоматом. Непосредственное описание конечного автомата с помощью диаграммы состояний UML может быть использовано для построения простого лексического анализатора [2].

Основная задача лексического анализа – разбить входной текст, состоящий из последовательности одиночных символов, на последовательности одиночных символов, на последовательности слов, или лексем, то есть выделить эти слова из непрерывной последовательности символов. Общая схема работы лексического анализатора такова:

- сначала выделяется отдельная лексема;
- ключевые слова распознаются явным выделением непосредственно из текста, либо сначала выделяется идентификатор, а затем делается проверка на принадлежность его множеству ключевых слов.

Библиографический список

1. Теория и реализация языков программирования: учебное пособие для СПО / В. А. Серебряков, М. П. Галочкин, Д. Р. Гончар, М. Г. Фуругян. — Саратов: Профобразование, 2021. — 372 с. — ISBN 978-5-4488-1013-8. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО ПроФобразование: [сайт]. — URL: <https://profspo.ru/books/102206> (дата обращения: 02.11.2022).

2. Леоненков, А. В. Самоучитель UML 2. / А. В. Леоненков. – СПб.: БХВ-Петербург, 2007. – 576 с.

ОВЕРЛЕЙНЫЕ КОМПЬЮТЕРНЫЕ СЕТИ

В.П. Корячко, А.А. Погудаев

Научный руководитель — В.П. Корячко

д.т.н., профессор

**Рязанский государственный радиотехнический университет
имени В. Ф. Уткина**

Оверлейной компьютерной сетью называют логическую сеть, развернутую поверх другой сети. На текущий момент существует огромное количество различных оверлейных сетей. Попробуем классифицировать оверлейные сети по ряду критериев, а также рассмотрим основные преимущества и недостатки оверлейных сетей.

Оверлейные сети могут использоваться для различных целей:

- создание аналога локальной сети, для совместной работы или развлечений;

- для обхода блокировок в сети интернет;

- для анонимизации;

- для обмена файлами;

- для организации доступа к компьютерам, которые находятся за NAT;

- для криптовалютных транзакций;

- для других нужд.

Все оверлейные сети по архитектуре можно разделить на несколько типов.