

Взаимодействие с интерфейсом Component Object Model на python

Вихляев Дмитрий Романович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье описывается принцип работы взаимодействия COM-объектов в операционной системе Windows. Реализация программы написана на языке программирования python. Результатом исследования является пример взаимодействия пользовательской программы с функциями MS Office Word.

Ключевые слова: COM, python, word.

Interaction with the Component Object Model interface in python

Vikhlyayev Dmitry Romanovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article describes the principle of operation of the interaction of COM objects in the Windows operating system. The implementation of the program is written in the python programming language. The result of the study is an example of the interaction of a user program with the functions of MS Office Word.

Keywords: COM, python, word.

1 Введение

1.1 Актуальность

COM является стандартом взаимодействия между компонентами в Windows. Множество приложений и служб в Windows используют COM для обеспечения взаимодействия между собой. COM позволяет взаимодействовать с другими приложениями через общие интерфейсы, тем самым упрощая взаимодействие между компонентами различных приложений. Благодаря высокому уровню контроля над вызовами и обменом данными обеспечивается возможность точно управлять тем, какие функции будут вызваны и как обработается результат. Интерфейс предоставляет доступ к разнообразным возможностям MS Office. Что позволяет создавать отчеты и дашборды в Excel на основе данных из внешних источников, проводить автоматизацию процессов отправки автоматических электронных писем, или создания и обновления презентаций, разрабатывать пользовательские интерфейсы для взаимодействия с Office-приложениями из других программ.

1.2 Обзор исследований

Л.А.Волкова, С.Г.Тагайцева провели исследование COM-соединения, выгрузки Excel через COM-объект Excel.application [1]. Р.В.Наумов в своей статье описал программирование python.com объекты в django [2]. В.И.Симашев, И.В.Викторов, И.В.Яковлев показали взаимодействие с объектами COM в исполняющей среде .NET [3]. А.А.Ланцев, С.П.Мочалов, С.Ю.Красноперов разработали систему распределенных вычислений на основе технологии COM [4]. А.В.Близнюк в своём исследовании создал и применил компоненты COM [5].

1.3 Цель исследования

Цель исследования – теоретическое и практическое описание возможностей COM-интерфейса, как способа автоматизации разработки программного обеспечения в операционной системе Windows.

2 Материалы и методы

Для реализации взаимодействия с COM-компонентами на языке python использована библиотека win32com, предоставляющая высокоуровневые методы работы с Windows API.

3 Результаты и обсуждения

COM (Component Object Model) - это стандартная технология в мире Windows, предназначенная для создания и взаимодействия с объектами в программном обеспечении. COM предоставляет стандартизированный способ создания, использования, обновления и удаления объектов в операционной системе Windows. Многие компоненты операционной системы сами являются COM-компонентами. Например, окна, диалоговые окна, файловые системы, принтеры и другие элементы интерфейса пользователя могут быть реализованы как COM-объекты. Что позволяет другим приложениям взаимодействовать с ними через стандартизированный интерфейс.

COM также позволяет создавать собственные компоненты, которые могут использоваться сторонними приложениями. Такой подход даёт возможность разработчикам реализовывать повторно используемые компоненты с определенным функционалом и предоставлять их другим приложениям. Возможность интегрировать различные технологии и языки программирования. Например, COM-объект, написанный на C++, может быть использован приложением, написанным на Visual Basic.

В COM объекты предоставляют функционал через интерфейсы. Интерфейс – это набор методов, которые объект обязуется реализовать. Объекты в COM могут поддерживать один или несколько интерфейсов. Например, объект COM для работы с файлами может предоставлять интерфейс с методами для открытия, чтения и записи файлов.

Каждый интерфейс в COM идентифицируется уникальным идентификатором GUID. Это обеспечивает уникальность идентификации интерфейса. COM-компоненты представляют собой исполняемые модули или

библиотеки, которые содержат объекты, реализующие определенные интерфейсы. Взаимодействие с объектами в COM осуществляется через вызов методов интерфейсов. Объекты COM могут предоставлять методы для выполнения операций или получения данных. Информация о зарегистрированных COM-компонентах хранится в реестре Windows. Это включает в себя пути к исполняемым файлам и GUID интерфейсов. Реестр предоставляет механизм для регистрации (и отмены регистрации) COM-объектов, определения их интерфейсов и другой связанной информации.

Работа с реестром позволяет операционной системе и приложениям узнавать, какие COM-компоненты доступны, и как ими пользоваться.

Информация о CLSID и ProgID хранится в реестре Windows. В частности, для 32-битных приложений имеется своя структура (HKEY_CLASSES_ROOT\CLSID, HKEY_CLASSES_ROOT).

Для 64-битных приложений используется аналогичная структура в разделах реестра HKEY_CLASSES_ROOT\Wow6432Node\CLSID и HKEY_CLASSES_ROOT\Wow6432Node.

Можно перейти к соответствующему разделу в реестре или использовать интерфейс командной строки, чтобы просмотреть доступные ключи COM-объектов, зарегистрированных в системе (рис.2).

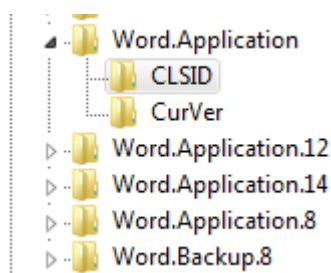


Рис. 1. Раздел компонента word в реестре

CLSID (Class Identifier) представляет собой уникальный идентификатор класса COM-объекта в системе Windows. Этот идентификатор представляется в реестре как строка символов формата (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx), где каждый x – это шестнадцатеричная цифра.

Первые 8 цифр (32 бита) – идентификатор производителя (Manufacturer Identifier, MID). Этот идентификатор выделяется Microsoft для каждого производителя COM-объекта.

Следующие 4 цифры (16 бит) номер версии объекта (Object Version Number, OVN). Определяет версию самого объекта.

Следующие 4 цифры (16 бит) Номер версии интерфейса (Interface Version Number, IVN). Определяет версию интерфейса объекта.

Последние 12 цифр (48 бит): Уникальный идентификатор объекта (Object Identifier, OID). Этот идентификатор выделяется производителем идентификации объектов.

Пример CLSID может быть 12345678-1234-5678-1234-56789ABCDEF0 (MID: 123456786 OVN: 12346 IVN: 56786 OID: 123456789ABCDEF0)

CLSID является частью системы идентификации COM-объектов и гарантирует их уникальность в системе. Когда приложение хочет создать экземпляр COM-объекта, оно использует его CLSID для обращения к системе и запроса соответствующего объекта.

Некоторые компоненты операционной системы Windows, такие как объекты работы, с реестром или объекты для работы с файлами, являются COM-компонентами. Для примера взята функция Microsoft Word, распознающая орфографические ошибки в словах и предлагающая список возможных исправлений (рис.2).

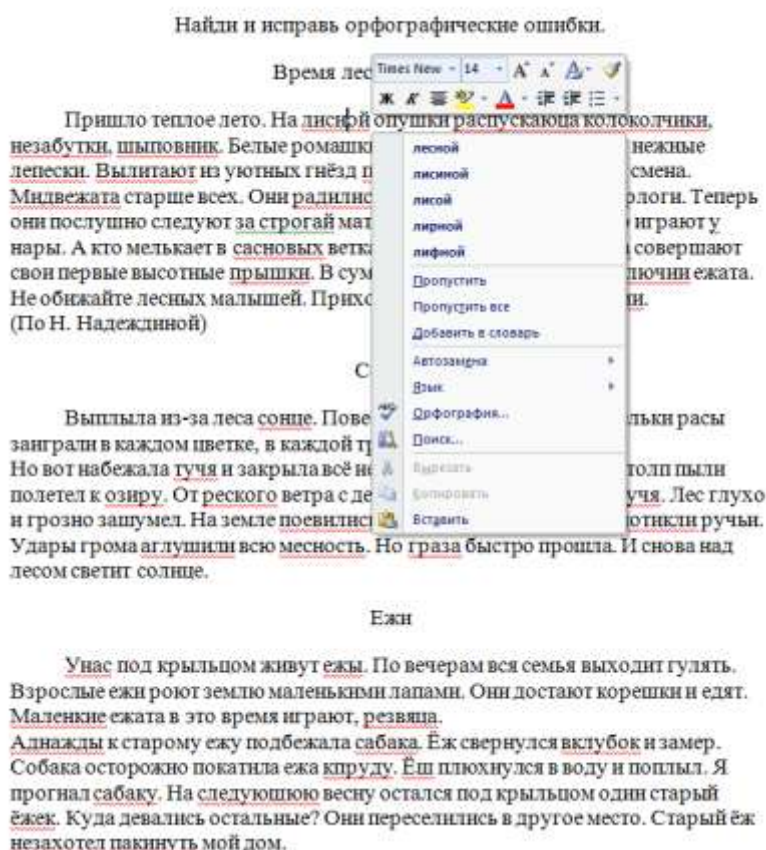


Рис. 2. Пример работы орфографических исправлений в word

«win32com» – это модуль для взаимодействия с COM-объектами в Windows, который позволяет Python взаимодействовать с приложениями и компонентами, предоставляемыми Windows API.

Модуль «win32com.client» предоставляет средства для работы с COM-объектами. С его помощью можно создавать, управлять и использовать объекты, предоставляемые COM-компонентами.

Модуль также предоставляет функционал для работы с системными сервисами через COM, такими как доступ, к реестру, управление службами Windows, возможность обработки событий от COM-объектов, что позволяет реагировать на изменения и события, происходящие в подключенных приложениях.

В модуле `pywin32`, `client` – это подмодуль, предоставляющий средства для работы с клиентскими COM-объектами в Python. Клиентские объекты COM обычно представляют собой объекты, предоставляемые другими приложениями или компонентами, которые требуется использовать из Python.

Основные функции модуля `win32com.client` включают в себя `Dispatch` и `Dynamic`. Эти функции позволяют создавать новые объекты COM. `Dispatch` используется для создания объектов на основе заранее определенных типов, в то время как `Dynamic` позволяет создавать объекты динамически, без заранее известных типов. В программе, приведённой на рисунке 3, создаётся объект MS Word. Из существующего документа извлекается список слов, которые `word` распознал как имеющие орфографические ошибки. Затем для каждого такого слова, вызывается метод генерирующий возможные исправления.

```
import win32com.client

def check_spelling_suggestions(document_path):
    try:
        # Создаем объект Word
        word_app = win32com.client.Dispatch("Word.Application")

        # Открываем документ Word
        word_doc = word_app.Documents.Open(document_path)

        # Устанавливаем диапазон, весь текст документа
        rng = word_doc.Range()

        # Получаем все слова с ошибками в документе
        misspelled_words = rng.SpellingErrors

        if misspelled_words.Count > 0:
            for word in misspelled_words:
                # Получаем предложенные исправления для каждого слова
                suggestions = word.GetSpellingSuggestions()

                print(f"Word: {word.Text}")
                if suggestions.Count > 0:
                    for i in range(1, suggestions.Count + 1):
                        suggestion_text = suggestions.Item(i).Name
                        print(f" Suggestion {i}: {suggestion_text}")
                else:
                    print(f"No spelling suggestions found for word")
            else:
                print("No misspelled words found in the document.")

    except Exception as e:
        print(f"An error occurred: {e}")
    finally:
        # Закрываем документ
        word_doc.Close()
```

Рис. 3. Использование орфографических функций `word` в программе на python

Результат работы программа выводит в консоль (рис. 4).

Word: лисной
 Suggestion 1: лесной
 Suggestion 2: лисиной
 Suggestion 3: лисой
 Suggestion 4: лирной
 Suggestion 5: лифной
 Suggestion 6: личной
 Suggestion 7: лианой
 Suggestion 8: Дисной
 Suggestion 9: Лыной
 Word: распускаяца
 No spelling suggestions found for word
 Word: колокольчики
 Suggestion 1: колокольчики
 Word: незабудки
 Suggestion 1: незабудки
 Word: шиповник
 Suggestion 1: шиповник
 Word: протягивают
 Suggestion 1: притягиваюе
 Suggestion 2: протягиваюе
 Word: сонцу
 Suggestion 1: сенцу
 Suggestion 2: синцу
 Suggestion 3: солнцу
 Suggestion 4: Солнцу
 Suggestion 5: гонцу
 Suggestion 6: донцу
 Suggestion 7: концу
 Suggestion 8: сонму
 Suggestion 9: сосцу
 Suggestion 10: Сонду
 Word: лепески
 Suggestion 1: лепестки
 Suggestion 2: лепешки
 Word: Вылетают
 Suggestion 1: Вылетают
 Suggestion 2: Вылкают
 Suggestion 3: Вычитают

Найди и исправь орфографические ошибки.

Время лесных малышей

Пришло теплое лето. На лисной опушке распускаяца колокольчики, незабудки, шиповник. Белые ромашки протягивают к солнцу свои нежные лепески. Вылетают из уютных гнезд птвички. У зверей взрастает смена. Миджежата старше всех. Они радиллис еще холодной зимой в берлоги. Теперь они послушно следуют за строгой матерью. Рыжие десята весело играют у нары. А кто мелькает в сасновых ветках? Да это дофкие бельчята совершают свои первые высотные прыжки. В сумерках выходят на охоту кюлочки ежата. Не обижайте лесных малышей. Приходите в лес верными друзьями. (По Н. Надеждиной)

Солнце

Выплыла из-за леса сонце. Повеселела лесная палына. Капельки расы заиграли в каждом цветке, в каждой травинке. Но вот набежала туча и закрыла всё небо. Загрустила природа. Столп пыли полетел к озеру. От реского ветра с деревьев посыпались сухие суча. Лес глухо и грозно зашумел. На земле появились мокрые пятна. С халмов потекли ручьи. Удары грома аглушили всю месность. Но граза быстро прошла. И снова над лесом светит солнце.

Ежи

Унас под крыльцом живут ежы. По вечерам вся семья выходит гулять. Взрослые ежи роют землю маленькими лапами. Они достают корешки и едят. Маленкие ежата в это время играют, ревяща. Аднажды к старому ежу подбежала сабака. Еж свернулся в клубок и замер. Собака осторожно покатила ежа кпруду. Еж плюхнулся в воду и поплыл. Я прогнал сабаку. На следующую весну остался под крыльцом один старый ежик. Куда девались остальные? Они переселились в другое место. Старый еж незахотел пакинуть мой дом.

Рис. 4. Вывод слов с орфографическими ошибками и возможные исправления

Таким образом, был проведён пример работы с интерфейсом СОМ-объекта на языке python. Использование функций и методов готовых программ значительно расширяет возможности процедур обычных библиотек. Грамотное проектирование приложения позволит использовать не только текстовую информацию, но и готовые графические решения.

Библиографический список

1. Волкова Л.А., Тагайцева С.Г. Использование СОМ-соединения, выгрузка excel через СОМ-объект excel.application // В сборнике: XI Всероссийский Фестиваль науки. Сборник докладов. Редколлегия: Д.Л. Щёголев, И.С. Соболев, Д.В. Монич, А.А. Смыков [и др.]. Нижний Новгород, 2021. С. 1168-1172.
2. Наумов Р.В. Программирование python.com объекты в django // Academy. 2016. № 12 (15). С. 11-12.
Симашев В.И., Викторов И.В., Яковлев И.В. Взаимодействие с объектами СОМ в исполняющей среде .NET // В сборнике: Актуальные проблемы общества, экономики и права в контексте глобальных вызовов. сборник материалов XX Международной научно-практической конференции. Санкт-Петербург, 2023. С. 19-23.
3. Мадьярова Г.А., Аубакиров Т.М., Куанышпаев М.К., Адамова К.А., Паримбек Ж.М. Сравнительный анализ онлайн-платформ // Вопросы

- устойчивого развития общества. 2021. № 11. С. 568-574.
4. Ланцев А.А., Мочалов С.П., Красноперов С.Ю. Разработка системы распределенных вычислений на основе технологии СОМ // В сборнике: Системы автоматизации в образовании, науке и производстве. AS'2001. Труды всероссийской научно-практической конференции. Под общей редакцией д.т.н., профессора С.М. Кулакова, д.т.н., профессора Л.П.Мышляева. 2001. С. 235-239.
 5. Близнюк А.В. Создание и применение компонентов СОМ // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2010. № 1. С. 117-128.
 6. Мазеин К.В., Поняшова А.С., Безносков П.П., Козырь А.А., Храмов С.В. Принципы взаимодействия систем 1с: предприятие через СОМ-соединение // Актуаль Аллея науки. 2017. Т. 1. № 12. С. 237-243.
 7. Коробко И. Создаём com-компоненты с помощью vbscript // Системный администратор. 2007. № 8 (57). С. 80-84.