

Тема магистерской диссертации:

Автоматизация сборки и
модификации разнородной
компьютерной документации:
разработка графического
интерфейса приложения и
формата файлов проекта.

Исполнитель: Утяшева И.А.

Научный руководитель: к.фм.н. Винокур В.В.

Задача

Предоставлен программный модуль, осуществляющий модификацию и конвертацию документации форматов СНМ и HTML.

Разработать:

1. Графический интерфейс для данного модуля.
2. Формат файлов проекта.

Проектирование интерфейса

1 этап:

Описание сценариев работы пользователей.

Одно из достоинств:

проблема юзабилити на ранней стадии изучена.

2 этап:

Построение протопов интерфейса:

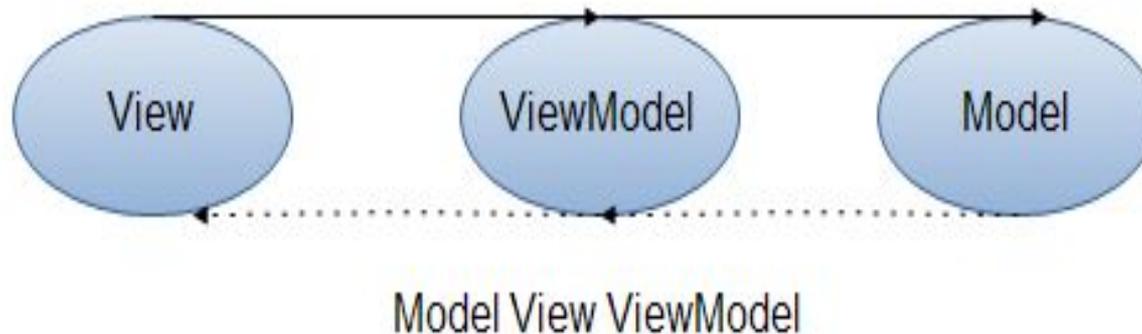
бумажные протипы и прототипы, построенные с помощью Microsoft Expression Blend + SketchFlow.

Реализация интерфейса

Выбран способ организации графического интерфейса приложений в **отдельных окнах (Single document interface)** в силу простоты реализации его и подходящей модели поведения.

Графический интерфейс построен на очень простом и одновременно очень мощном паттерне проектирования клиентских приложений **DataModel-View-ViewModel**.

Model View ViewModel (MVVM)



Модель (Model) отвечает за бизнес-логику и управление состоянием объекта.

Представление (View) отвечает только за отрисовку элементов графического интерфейса.

ViewModel не содержит ссылок на вид:

связывания между View и ViewModel становятся возможными потому, что экземпляр ViewModel задается в View как DataContext.

Самый важный момент WPF, делающий MVVM очень удобным шаблоном – это **инфраструктура привязки данных:**

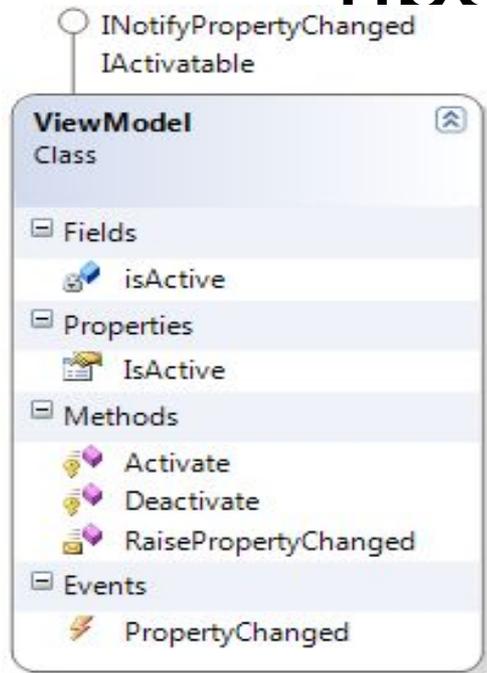
за счет привязки свойств представления к модели представления получается слабое связывание этих компонентов

MVVM: оповещение

Когда какое-либо свойство модели изменяется: новое значение автоматически передается в представление через **механизм связывания данных**.

Модель должна реализовывать интерфейсы **INotifyPropertyChanged** или **INotifyCollectionChanged**.

Базовый класс для представлений



Все наследуемые классы представляют собой отдельные представление.

Благодаря наследуемому интерфейсу `INotifyPropertyChanged` представления оповещают модели данных об изменениях и событиях.

`IActivatable` позволяет представлению задавать поведение, когда оно добавляется и удаляется из визуального дерева.

Каналы связи представлений

С помощью интерфейса IMessageBroker и его реализацией MessageBroker представления могут обмениваться сообщениями между собой. Метод GetChannel позволяет строить каналы связи между представлениями.

