# Тема проекта

Здравствуйте уважаемая комиссия, я Пронин А.С. из группы ИУ7-82б, и тема моей работы “Метод анализа активности пользователей системы автоматизированного проектирования (САПР) с использованием поиска последовательных шаблонов”.

# Актуальность

Анализ активности пользователей позволяет разработчикам понять, как они взаимодействуют с системой и внести улучшения в интерфейс или функциональность, для повышения эффективности использования системы.

Кроме этого, на основе полученной информации можно сделать выводы на каких аспектах системы стоит сосредоточить усилия.

А также, выявить аномальное или нежелательное поведение, которое может указывать на возможные проблемы взаимодействия пользователя с интерфейсом или ошибки в системе.

Уровень удобства использования программного интерфейса влияет на качество всего ПО в целом. Признаком недостаточного уровня удобства использования является наличие проблем взаимодействия пользователя с интерфейсом. Они могут быть связаны либо со сложностью принятия решений, либо с непониманием ответа системы.

Проблемы взаимодействия в большинстве случаев можно определить по наличию в данных активности пользователей шаблонов.

Для их обнаружения применяются различные методы анализа собираемых данных – как требующие ручного анализа (например, тепловые карты), так и использующие алгоритмы автоматического анализа. Автоматический анализ экономит время и деньги, так как эксперты вместо анализа всех данных фокусируют внимание на отдельных областях пользовательского интерфейса, где были выявлены соответствующие шаблоны.

# Цель и задачи

На следующем слайде представлены цель и задачи.

# Сравнение рассмотренных методов

В ходе работы мною были рассмотрены различные методы анализа активности пользователей, сравнение которых представлено в таблице на слайде.

Поскольку активность пользователей представляет собой последовательность действий и их характеристик, производящихся в определенный момент времени, то за основу был выбран алгоритм GSP, т.к. он учитывает время совершения транзакций.

# Формализованная постановка задачи

Задача анализа активности пользователей САПР представлена в нотации IDEF0, которую Вы можете видеть на слайде. На вход программе подаются информация о выполненных командах и пользовательские параметры: минимальный уровень поддержки, минимальный и максимальный разрывы между командами в секундах. Ограничения на входные данные представлены на экране

Используя методы поиска последовательных шаблонов, система определяет часто встречающиеся последовательности команд, их уровень поддержки и коэффициент зависимости.

# Характеристики последовательностей

Значение поддержки последовательности равно проценту сессий, которые ее поддерживают.

Сессия поддерживает последовательность, если содержит все ее элементы в том же порядке и временные промежутки между каждыми двумя соседними командами удовлетворяют параметрам, заданным пользователем.

Коэффициент зависимости показывает, насколько команды в последовательности зависят друг от друга и считается как отношение поддержки последовательности к произведению поддержек всех подпоследовательностей, состоящих из 1 команды. Если значение коэффициента <= 1, значит зависимости нету. Если же > 1, то зависимость есть. Чем больше единицы, тем вероятней то, что эти команды использовались вместе.

# Уровень поддержки

Значение поддержки последовательности равно проценту сессий, которые ее поддерживают.

Сессия поддерживает последовательность, если содержит все ее элементы в том же порядке и временные промежутки между каждыми двумя соседними командами удовлетворяют параметрам, заданным пользователем.

На слайде представлен пример сессии и последовательностей, которые она поддерживает или нет, где min\_gap и max\_gap это минимальный и максимальный разрыв, задаваемый пользователем.

# Коэффициент зависимости

Коэффициент зависимости показывает, насколько команды в последовательности зависят друг от друга и считается как отношение поддержки последовательности к произведению поддержек всех подпоследовательностей, состоящих из 1 команды. Если значение коэффициента <= 1, значит зависимости нету. Если же > 1, то зависимость есть. Чем больше единицы, тем вероятней то, что эти команды использовались вместе.

На слайде представлен пример расчета коэффициента зависимости.

# Ключевые этапы алгоритма

Из особенностей предлагаемого метода можно выделить следующие:

Перед обработкой данных алгоритмом, они преобразовываются в таблицу базы данных.

Элемент последовательности может состоять только из одной команды. Следовательно, нет необходимости в использовании скользящего окна.

Разработанный метод состоит из двух основных этапов: Генерация кандидатов и Подсчет поддержки кандидатов. При этом на первом шаге алгоритма берутся все возможные одноэлементные последовательности.

На следующих слайдах представлены схемы для данных этапов.

# Генерация кандидатов

При генерации кандидатов просто перебираются все сочетания текущих последовательностей, и если условие их объединения выполняется, то добавляется новый кандидат.

# Проверка поддержки кандидата сессией

При подсчете поддержки кандидатов каждая сессия проверяется на содержание рассматриваемой последовательности. На данном этапе алгоритм переключается между двумя фазами: поиск следующей и предыдущей команды (по англ. forward и backward phase). Происходит это пока последовательность не будет полностью найдена или же какая-либо команда из нее не будет отсутствовать.

# Структура программного обеспечения

На данном слайде представлена структура ПО.

Процесс взаимодействия с программной обычно выглядит следующим образом:

Модуль преобразования логов записывает данные в таблицу с помощью модуля взаимодействия с базами данных. А модуль вычисления часто встречающихся последовательностей, реализующий разработанный метод, на основе этой базы данных вычисляет результат, для передачи его пользователю через интерфейс.

# Сравнительный анализ времени выполнения метода в зависимости от параметров

В качестве исследования мною был проведён сравнительный анализ времени выполнения разработанного метода в зависимости от параметров минимальной поддержки и минимального, максимального разрывов между командами.

Как и ожидалось, при уменьшении минимального уровня поддержки, время выполнения будет расти т.к. в таком случае больше последовательностей будут проходить отбор, что видно по 1ому графику. Особенно видна разница между временем выполнения при значениях минимальной поддержки 0.02 и 0.01.

На 2ом графике мы видим, что при увеличении минимального разрыва между командами, время выполнения уменьшается т.к. в таком случае получается меньше последовательностей из-за увеличения ограничения. К тому же, чем больше записей в базе данных, тем сильнее влияет изменения параметра min\_gap.

На последнем графике видно, что при уменьшении максимального разрыва между командами, время выполнения тоже уменьшается, потому что в этой ситуации, также получается меньше последовательностей проходят ограничения.

# Сравнительный анализ времени выполнения этапов метода

Также мною был проведён сравнительный анализ времени выполнения этапов метода, в результате которого, можно сделать вывод, что подсчет поддержки кандидатов занимает большую часть времени, чем их генерация.

# Заключение

По итогу проделанной работы была достигнута цель и решены все поставленные задачи, перечисленные на экране.

По итогу проделанной работы была достигнута цель – разработан и программно реализован метод анализа активности пользователей САПР с использованием поиска последовательных шаблонов, а также были решены все поставленные задачи, перечисленные на экране.

# Перспективы дальнейшего развития

В качестве возможных путей развития данной разработки можно выделить следующие:

* Возможность предсказания следующей команды пользователю
* Подсчет процента содержания найденных последовательностей в сессиях или других характеристик для получения дополнительной информации
* Оценка времени, необходимого для выполнения последовательности

# Демонстрация

Спасибо за внимание, теперь можно приступить к вопросам и демонстрации ПО.