УДК 004.42

Давыдов Н.В.,
студент магистратуры,
2 курс, факультет «Космический»
МФ МГТУ им. Н.Э. Баумана
Россия, г. Мытищи

ОЦЕНКА ПОДХОДОВ К ПОСТРОЕНИЮ СИСТЕМЫ ОГРАНИЧЕНИЙ ДЕЙСТВИЙ В СЛОЖНЫХ ПРОГРАММНЫХ РЕШЕНИЯХ НА ОСНОВЕ ЭКСПЕРТНОГО ОПРОСА РАЗРАБОТЧИКОВ

Аннотация: *В статье рассматриваются подходы к ограничению действий в зависимости от текущего состояния приложения. Такие механизмы позволяют предотвращать недопустимые или конфликтующие операции, например обращение к неинициализированному ресурсу. Цель исследования — сравнить различные подходы по численным показателям и упростить выбор архитектурного решения с учетом требований проекта.*

Ключевые слова: *система разрешений, конечный автомат, централизованный менеджер ограничений, флаговая модель, локальные условные проверки, промышленное программное обеспечение, экспертный опрос.*

Abstract: *The article examines approaches to restricting actions depending on the current state of an application. Such mechanisms help prevent invalid or conflicting operations, such as accessing an uninitialized resource. The aim of the study is to compare different approaches using quantitative indicators and to simplify the choice of an architectural solution taking into account project requirements.*

Keywords: *permission system, finite state machine, centralized constraint manager, flag-based model, local conditional checks, industrial software, expert survey.*

Для формирования экспертной оценки было произведено анкетирование разработчиков, работа которых связана непосредственно с построением приложений с описанной проблематикой. Стоит отметить, что разработчики имеют различный уровень квалификации, что определяется их должностью, которая учитывается при анкетировании. Респондентам в рамках анкетирования предлагается оценить описываемые подходы по ряду инженерных критериев.

Результаты опроса после их анализа позволяют сформировать систему численных показателей в формате таблицы, которая позволит лицу принимающему решение об архитектуре быстро выбрать наиболее подходящее архитектурное решение.

Локальные условные проверки представляют собой реализацию, при которой проверки выполняются с помощью условных операторов отдельно под каждое действие. Когда пользователь собирается выполнить какое-то действие, выполняется набор проверок и по их результатам принимается решение о допустимости или недопустимости действия. Из недостатков такого подхода можно выделить [1, 2, 3]:

- Необходимость получать из места проверки, как правило из обработчика события, доступ к большому количеству элементов приложения для их опроса;
- Сложность в добавлении новых проверок в каждый обработчик;
- Низкая информативность подхода.

Из преимуществ можно выделить [1, 2, 3]:

- Быстрая и легкая реализация;

- В небольших проектах без перспективы расширения может оказаться более надежным решением, благодаря более простому процессу проверки работоспособности относительно других решений.

Флаговая модель может рассматриваться как упрощённый вариант менеджера ограничений или конечного автомата в зависимости от реализации. Ее смысл в том, чтобы описывать состояние приложения некоторым количеством глобальных флагов. Это может быть реализовано и в формате отдельного флага на каждое действие, и в формате отдельных флагов для каждого состояния, и в промежуточных форматах. Из недостатков такого подхода можно выделить [1, 2, 3]:

- Потенциальные конфликты флагов;
- Информативность выше, чем при условной логике, но все еще сложно проследить механизмы изменения состояния флагов в больших проектах;
- Сложность в масштабировании приложения при появлении новых флагов.

Из преимуществ можно выделить [1, 2, 3]:

- Быстрая и легкая реализация;
- Информативность выше, чем при условной логике;
- Проверки не требуют опроса большого количества элементов приложения.

Централизованный менеджер ограничений представляет собой отдельный модуль приложения, который берет на себя задачу оценки допустимости действий. При этом сам механизм проверки может быть реализован по-разному — опрос элементов приложения, запоминание их состояния, или даже построение машины состояний. Данный подход решает одну из ключевых проблем — необходимость описывать логику в каждом обработчике. В случае применения централизованного менеджера ограничений необходимо только запросить у него текущий набор правил и

проверить нужное ограничение. Из недостатков такого подхода стоит отметить [1, 3, 4]:

- Реализация данного подхода требует больше времени и навыков;
- Первичная реализация и настройка тестовых сценариев требуют больше времени, чем в случае локальных проверок;
- Внутренняя логика построения ограничений может быть сложной;
- Построение правил может слишком точечно анализировать состояние элементов приложения, что снижает уровень абстракции.

Из преимуществ такого подхода можно выделить [1, 3, 4]:

- Правильная реализация позволяет надежно исключить конфликты ограничений;
- Хорошая возможность масштабирования;
- Качественная внутренняя логика дает хорошую информативность о работе системы ограничений;
- Возможность учитывать состояние отдельных элементов.

Конечный автомат представляет собой набор состояний и допустимых переходов между ними. При получении какого-либо входного воздействия автомат, если это допустимо, выполнит переход в новое состояние. Таким образом система ограничений будет перестроена уже под новое состояние. Из недостатков данного подхода стоит выделить [1, 3, 4]:

- Реализация данного подхода требует более высокой квалификации разработчиков;
- Крайне желательно иметь представление о наборе состояний на начальном этапе разработки;
- Состояния являются абстракциями высокого уровня, поэтому точечные ограничения, зависящие от отдельных параметров контекста, требуют дополнительных проверок.

Из преимуществ данного подхода можно выделить [1, 3, 4]:

- Высокая абстракция дает очень высокую информативность о работе ограничений;
- Отладка проще, чем у других рассмотренных подходов;
- Крайне высокая поддержка масштабируемости.

Для оценки эффективности рассмотренных ранее подходов предлагается использовать метод экспертного опроса. В качестве экспертов привлекаются разработчики промышленного предприятия, которые в работе постоянно решают задачи построения устойчивых и надежных систем ограничений. Категории специалистов и их обозначения в анкете приведены в таблице 1.

Таблица 1.

Категории специалистов

Категория респондента	**Обозначение в анкете**
Инженер-программист III категории	ИП III
Инженер-программист II категории	ИП II
Инженер-программист I категории	ИП I
Ведущий инженер-программист	ВИП

Опрос проводится анонимно, чтобы снизить предвзятость оценок.

Критерии оценки для сравнительного анализа приведены в таблице 2.

Таблица 2.

Описание критериев сравнительного анализа

Критерий	**Описание**	**Обозначение**
Сложность разработки	Насколько сложно реализовать подход в проекте	Сл.Р.
Время первичной реализации	Оценка трудозатрат на начальное внедрение подхода	Т реал.
Вероятность ошибок в коде	Насколько высок риск появления дефектов из-за выбранного подхода	Р ош.
Сложность сопровождения	Насколько сложно изменять и поддерживать систему при развитии проекта	Сл.Сопр.
Риск дублирования	Вероятность повторения одинаковых	Д

Критерий	Описание	Обозначение
логики	проверок в разных частях приложения	
Масштабируемость	Насколько хорошо подход работает при росте количества состояний, команд и сценариев	М
Тестируемость	Насколько удобно проверять корректность разрешений и ограничений	Т
Прозрачность архитектуры	Насколько легко понять и объяснить логику разрешений в рамках подхода	П

Шкала оценки приведена в таблице 3.

Таблица 3.

Интерпретация оценок

Балл	Интерпретация
1	Низкий уровень
2	Ниже среднего
3	Средний уровень
4	Высокий уровень
5	Очень высокий уровень

Анкета экспертного опроса приведена в таблице 4.

Таблица 4.

Анкета опроса

Должность								
Подход	**Сл.Р.**	**Т реал.**	**Р ош.**	**Сл.Сопр.**	**Д**	**М**	**Т**	**П**
Локальные условные проверки								
Централизованный менеджер ограничений								

Должность								
Флаговая модель								
Конечный автомат								
Наиболее оптимальный вариант								

После проведения опроса были рассчитаны следующие показатели:

- Среднее арифметическое значение;
- Медиана;
- Минимальное значение;
- Максимальное значение;
- Стандартное отклонение;
- Количество ответов;
- Гистограмма распределения ответов (для качественного вопроса).

Поскольку значения стандартного отклонения по большинству критериев не превышают 0,7 балла, результаты представлены в формате среднего арифметического значения и стандартного отклонения в таблицах 5 и 6. Результаты опроса выделены в две группы. Так как разработчик III категории всего один он был объединен с разработчиками II категории. Ведущие разработчики и разработчики I категории были объединены в одну группу, так как все опрашиваемые из этих категорий имеют опыт разработки от 9 лет, что более чем достаточно для высокого уровня понимания архитектуры построения приложений. Таким образом было получено две группы опрашиваемых – начинающие и опытные разработчики.

Таблица 5.

Результаты опроса для ИП III (16 % - 1 чел.) и ИП II (84% - 5 чел.)

Подход	Сл.Р.	Т реал.	Р ош.	Сл. Сопр.	Д	М	Т	П
Локальные условные проверки	1,6±0,5	1,5±0,5	4,1±0,7	4,0±0,7	4,3±0,6	2,0±0,6	2,1±0,5	2,2±0,6
Централизованный менеджер ограничений	3,0±0,6	2,9±0,6	2,8±0,6	2,7±0,5	2,3±0,5	3,4±0,6	3,5±0,6	3,3±0,6
Флаговая модель	2,3±0,6	2,1±0,5	3,8±0,7	3,7±0,7	3,5±0,6	2,7±0,6	2,6±0,6	2,4±0,6
Конечный автомат	4,1±0,7	4,0±0,7	2,0±0,5	2,2±0,6	1,8±0,5	4,2±0,6	4,0±0,6	4,1±0,6
Наиболее оптимальный вариант	Конечный автомат							

Исходя из полученных результатов от группы начинающих разработчиков можно сделать вывод, что они склонны в положительном ключе оценивать более простые в реализации архитектуры, но при этом четко понимают, что использование таких подходов может привести к потенциальным проблемам в будущем. Наиболее оптимальным вариантом они склонны считать конечный автомат, который успешно сочетает в себе умеренную сложность в реализации, но при этом обеспечивает хорошие показатели масштабируемости и прозрачности полученной системы.

Таблица 6.

Результаты опроса для ИП I (60 % - 6 чел.) и ВИП (40 % - 4 чел.)

Подход	Сл.Р.	Т реал.	Р ош.	Сл. Сопр.	Д	М	Т	П
Локальные условные проверки	1,4±0,4	1,3±0,4	4,5±0,5	4,4±0,5	4,6±0,4	1,8±0,5	1,9±0,5	1,9±0,4

Подход	Сл.Р.	Т реал.	Р ош.	Сл. Сопр.	Д	М	Т	П
Централизованный менеджер ограничений	2,7±0,5	2,6±0,5	2,5±0,5	2,5±0,5	2,1±0,4	3,6±0,5	3,8±0,5	3,6±0,5
Флаговая модель	2,1±0,5	2,0±0,5	4,2±0,6	4,1±0,6	3,9±0,5	2,3±0,5	2,4±0,5	2,2±0,5
Конечный автомат	3,6±0,6	3,5±0,6	1,6±0,4	1,8±0,4	1,5±0,4	4,6±0,4	4,5±0,4	4,5±0,4
Наиболее оптимальный вариант	Конечный автомат							

Гистограмма распределения ответов на качественный вопрос о наиболее оптимальном подходе представлена на рисунках 1 и 2.

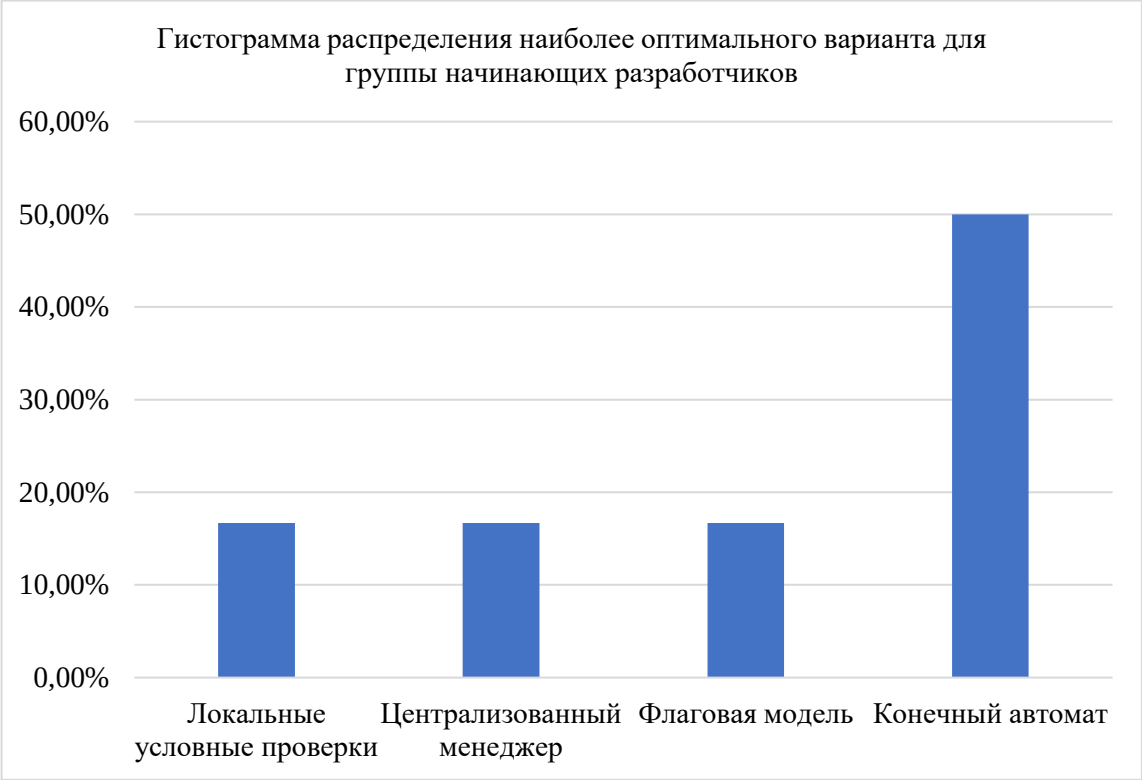

Рисунок 1. Гистограмма распределения наиболее оптимального варианта для группы начинающих разработчиков

Рисунок 2. Гистограмма распределения наиболее оптимального варианта для группы опытных разработчиков

В результате анализа полученных результатов можно сделать вывод, что одним из наиболее оптимальных вариантов разработчики выбирают построение конечного автомата. Также с ростом квалификации специалистов наблюдается тренд на снижение оценок сложности реализации описываемых подходов, но в целом все опрашиваемые респонденты демонстрируют примерно одинаковую позицию по предложенному списку вопросов.

В результате данной работы были исследованы подходы к построению систем контроля ограничений и построена сравнительная таблица для помощи в выборе наиболее подходящего решения под конкретный проект. Также на основе опроса разработчиков был сделан вывод, что наиболее оптимальной системой зачастую оказывается конечный автомат.

Использованные источники:

1. Бегджанов Б. Г. Оценка надежности программного обеспечения информационных систем // Вестник науки. 2024. №5 (74). URL: https://cyberleninka.ru/article/n/otsenka-nadezhnosti-programmnogo-obespecheniya-informatsionnyh-sistem-1 (дата обращения: 07.04.2026).

2. Кокурин Е. А. Исследование и разработка архитектуры информационной системы анализа использования программного обеспечения предприятия // Международный журнал гуманитарных и естественных наук. 2022. №11-2. URL: https://cyberleninka.ru/article/n/issledovanie-i-razrabotka-arhitektury-informatsionnoy-sistemy-analiza-ispolzovaniya-programmnogo-obespecheniya-predpriyatiya (дата обращения: 29.03.2026).

3. Яровая Екатерина Владимировна Принципы построения архитектуры программного обеспечения // E-Scio. 2022. №8 (71). URL: https://cyberleninka.ru/article/n/printsipy-postroeniya-arhitektury-programmnogo-obespecheniya (дата обращения: 15.04.2026).

4. Нурутдинов Тимур Айратович Архитектура и программное обеспечение для сбора пользовательских событий (CLICKSTREAM) и их последующего анализа // Universum: технические науки. 2023. №12-1 (117). URL: https://cyberleninka.ru/article/n/arhitektura-i-programmnoe-obespecheniya-dlya-sbora-polzovatelskih-sobytiy-clickstream-i-ih-posleduyuschego-analiza (дата обращения: 15.04.2026).