

ІНФОРМАТИКА, КОМП'ЮТЕРНА ТЕХНІКА

УДК 004.91

**ЗАСТОСУВАННЯ СКІНЧЕНИХ АВТОМАТІВ ДЛЯ ПОБУДОВИ СИСТЕМИ
ДОКУМЕНТООБІГУ НА ПРИКЛАДІ SPRING STATEMACHINE**

А. Джоші

студент 5 курсу, група КНІТ-51м, навчально-науковий інститут автоматичної, кібернетики та
обчислювальної техніки

Науковий керівник – д.т.н., професор П. М. Мартинюк

*Національний університет водного господарства та природокористування,
м. Рівне, Україна*

В статті проведено огляд технології Spring Statemachine та наведено можливості її використання для побудови системи документообігу в сфері іпотечного бізнесу.

Ключові слова: скінчений автомат, стани, переходи, фреймворк.

В статье проведен обзор технологии Spring Statemachine и представлены возможности ее использования для построения системы документооборота в сфере ипотечного бизнеса.

Ключевые слова: конечный автомат, состояния, переходы, фреймворк.

In the article review of Spring Statemachine technology has been done, described use cases for building documents workflow system in mortgage business sphere.

Keywords: finite state-machine, states, transitions, framework.

В теперішній час фінансові іпотечні організації все частіше зіштовхуються з проблемою управління справами їх клієнтів, основною причиною якої є ускладнення документообігу. В межах однієї справи може існувати декілька різних процедур обробки тих чи інших документів – від простої технічної перевірки на коректність введених даних до індивідуальної перевірки відповідності отриманих документів заданим правилам. Слід звернути увагу, що процес надання іпотеки та її обробки конкретних документів, що з цим пов'язані, не може бути повністю автоматизованим. Система обробки документів повинна дозволяти працівникам установи вирішувати питання їх клієнтів в індивідуальному порядку (з огляду на приватні особливості клієнта – тимчасова непрацездатність, сімейні обставини тощо), але в той же час автоматизувати сценарії, що є загальними для кожної справи (автоматична відправка загальних положень тощо).

Зрозуміло, що написання коду такої системи в імперативному стилі є досить складною задачею. Перенесення бізнес-правил в код в даній парадигмі може супроводжуватися великою кількістю рудиментарних ускладнень та складністю підтримки. Натомість, використовуючи декларативний підхід, є можливість реалізувати необхідний функціонал, не витрачаючи великих ресурсів часу. Також відомо, що код написаний в декларативному стилі є більш зручним для читання та супроводу.

Аналіз останніх досліджень. Для вирішення проблеми побудови скінченого автомата часто використовується імперативний підхід з використанням умовних операторів та періодичних опитувань автомата на предмет змін, проте даний варіант є досить складним у супроводі та процесі кодування. Spring Statemachine є частиною фреймворку Spring, основною метою якої є створення скінченого автомата та управління переходами між його станами. Опис станів та подій, додаткових користувацьких функцій та перевірок виконується

в декларативному стилі, що забезпечує зручність в розробці та підтримці системи.

Методика досліджень. Для вирішення проблеми побудови скінченного автомата запропоновано використання Spring Statemachine як технології, здатної до декларативного опису станів та переходів автомата з додатковими можливостями конфігурації, покликані спростити розробку подібних систем.

Постановка завдання Для проведення аналізу доцільності застосування скінчених автоматів для побудови системи документообігу було поставлено наступні завдання: огляд технології Spring Statemachine та опис можливостей технології для побудови системи документообігу в сфері іпотечного бізнесу.

Результати досліджень. Технологія Spring Statemachine виявилася зручним та потужним інструментом для побудови скінченого автомата системи документообігу, що дає можливість декларативного опису, простого розширення існуючого функціоналу та його підтримки.

Скінченим автоматом називають п'ятірку значень [1]:

$$M = (Q, \Sigma, \delta, q_0, F) \quad (1)$$

де Q – скінчена множина станів автомата, $Q = \{q_0, q_1, \dots, q_n\}$;

Σ – скінчена множина вхідних символів $\Sigma = \{a_1, a_2, \dots, a_n\}$;

δ – функція переходу скінченого автомата, $Q \times \Sigma \rightarrow P(Q)$;

$q_0 \in Q$ – початковий стан автомата;

$F \subseteq Q$ – множина заключних станів автомата.

Виникнення та розвиток теорії автоматів тісно пов'язані з розвитком обчислювальної техніки та розробкою моделей процесів обробки інформації в складних системах.

Spring Statemachine є реалізація скінчених автоматів в екосистемі Spring, основним призначенням якої є скінчені автомати різної складності, використання тригерів, перевірок переходів, інтеграція з контекстом застосунку та можливість збереження стану автомата до зовнішнього ресурсу [2].

Основними сценаріями використання технології є:

- застосунки або модулі застосунку, що можуть бути представлені у вигляді станів;
- поділ складної логіки на простіші керовані задачі;
- використання асинхронних операцій.

Множина станів та подій автомата Spring Statemachine задається за допомогою переліку (enum), а конфігурація переходів та перевірок за допомогою низки конфігураційних методів. Для використання автомата необхідно оголосити його та підв'язати (autowire) за допомогою стандартних анотацій контексту Spring.

Spring Statemachine дозволяє отримати доступ до контексту автомата в будь-який момент часу. Контекст містить у собі поточний стан, дозволені переходи, службові заголовки та розширений стан (extended state), що дозволяє зберігати дані будь-якого характеру у контексті, в свою чергу, дозволяючи використовувати перевірки переходів.

Під системою документообігу розуміється сукупність методів, засобів і персоналу, що підтримує документообіг у межах встановленого регламенту. Система електронного документообігу – організаційно-технічна система, що забезпечує процес створення, управління доступом і поширення електронних документів у комп'ютерних мережах, а також контроль над потоками документів в організації [3].

До основних принципів електронного документообігу належать: однократна реєстрація документа в цілях його однозначної ідентифікації; можливість паралельного виконання операцій; безперервність руху документа; єдина база документної інформації; організована система пошуку документа; система звітності за різними статусами і атрибутами документа.

Розглянемо схему системи документообігу, представлену на рисунку.

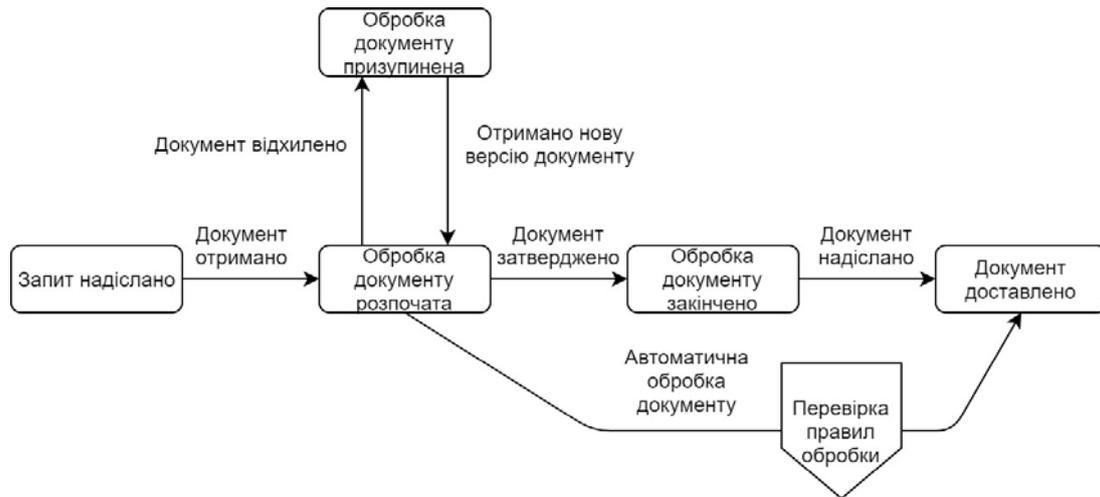


Рисунок. Система документообігу іпотечної справи

Для вирішення задачі створення системи документообігу було використано наступні стани: MP_DOCUMENT_REQUESTED, SP_DOCUMENT_RECEIVED, SP_DOCUMENT_APPROVED, SP_DOCUMENT_SENT, SP_DOCUMENT_DECLINED. Зв'язок між станами забезпечено за допомогою наступних переходів: TP_DOCUMENT_RECEIVED, SP_DOCUMENT_REJECTED, SP_DOCUMENT_APPROVED, SP_DX_SENT. Важливо звернути увагу на останній перехід, так як він може виконуватися як із стану ручної перевірки документу, так і автоматично. Така поведінка забезпечується використанням перевірок переходів (guards).

Оскільки створення контексту автомата є ресурсоємною задачею існує можливість використовувати один і той же автомат для всіх документів. Оскільки стани документів відрізняються один від одного, постає питання ручного переведення автомата у стан, що відповідає документові. Для цього зазвичай використовують відновлювачі станів (persist resolvers). Основною метою даного рівня автомата є відтворення контексту автомата із будь-яких джерел – файлу, бази даних або віддаленого сервісу. Слід звернути увагу, що використання єдиного автомата для обробки всіх документів вимагає особливого контролю за багатопотоковістю – відтворення контексту та обробка запитів повинні виконуватися в синхронізованому режимі. Якщо ж питання продуктивності є пріоритетним, потрібно розглянути можливість створення пулу скінчених автоматів.

В ході використання Spring Statemachine було значно спрощено створення системи документообігу в застосунку. Логіку документообігу було повністю винесено в контекст автомата, що унеможливило вплив цього модулю на інші частини застосунку. Завдяки декларативному підходу у опису автомата було покращено зручність підтримки даної системи та зменшено вплив людського фактора на загальну якість застосунку. Реалізація нових вимог не потребувала великої кількості змін у застосунку, а лише розширювала існуюче рішення.

Отже, Spring Statemachine може стати доцільною технологією для побудови скінченого автомата системи документообігу в прикладних застосунках. В умовах частих змін вимог до продукту функціонал даної технології допомагає розширювати існуючу логіку застосунку, але не вносити значних архітектурних змін в нього.

Список використаних джерел:

1. Енциклопедія кібернетики / відпов. ред. Глушков В. М. – Т. 1 (А-Л). – К. : Головна редакція УРЕ АН УРСР, 1973. – 584 с.
2. Документація Spring Statemachine [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <https://docs.spring.io/spring-statemachine/docs/2.0.0.RELEASE/reference/htmlsingle/>
3. Кукарін О. Б. Електронний документообіг та захист інформації / О. Б. Кукарін. – К. : НАДУ, 2015. – 84 с.