

Олексій Джусь¹, Михайло Лобур²

¹Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери 12, Львів, Україна, E-mail: oleksii.p.dzhus@lpnu.ua

²Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери 12, Львів, Україна, E-mail: mykhaylo.v.lobur@lpnu.ua, ORCID 0000-0001-7516-1093

ВІД АВІАЦІЙНИХ ДО ВЕБ СТАНДАРТІВ: ТРАНСФОРМАЦІЯ SSIM-ФАЙЛІВ В JSON ФОРМАТ

Отримано: березень 01, 2024 / Переглянуто: квітень 01, 2024 / Прийнято: квітень 05, 2024

© Джусь О., Лобур М., 2024

<https://doi.org/>

Анотація. У зв'язку з динамічними змінами в авіаційному секторі успішний обмін та інтеграція даних планування відіграють вирішальну роль у забезпеченні збереження операційної ефективності авіаліній. Незважаючи на те, що SSIM (Standard Schedules Information Manual) – файл вважається надійним форматом, він має деякі обмеження щодо взаємодії та сумісності з іншими веб-технологіями та програмами через його текстовий формат строго визначений специфікацією. У цій роботі ми пропонуємо підхід до перетворення SSIM-файлів у формат JSON (JavaScript Object Notation).

Одним із центральних аспектів цієї роботи є детальна модель перетворення, яка враховує всі важливі структури та рівні інформації SSIM-файлу, проводить їх синтаксичний аналіз і перетворює їх в гнучкий і легко відновлюваний формат JSON. Для перетворення набору авіа рейсів у поєднану лінію польотів використовується модифікований метод гілок і меж. За допомогою низки прикладів та експериментів стаття має на меті показати ефективність даної методології з точки зору збереження цілісності даних, забезпечення їх точності. Зокрема, розглядаються проблеми, з якими стикаються під час перетворення, наприклад, як врахувати узгодженість даних рядків із фіксованою шириною у SSIM-файлі. Перехід від орієнтованих на авіацію форматів, як-от SSIM-файл, до глобальних веб-форматів, як-от JSON, – це процес, який виходить за межі технологій і має на меті збільшення гнучкості та зручності співпраці, а також обробки даних в авіаційній галузі. Це дослідження допомагає впровадити цю зміну, пропонуючи простий, ефективний і масштабований підхід для перетворення розкладів авіакомпаній у формат, який за своєю суттю відповідає потребам сучасних інформаційних додатків та систем.

Ключові слова: SSIM, JSON, авіалінії, формат даних, синтаксичний аналіз, метод гілок і меж

Вступ

Використання сучасних інформаційних технологій та стандартів даних в авіаційній галузі може підвищити операційну ефективність, зменшити вартість авіа-перельотів і покращити якість обслуговування[1]. Сучасні технології, такі як автоматизація, Інтернет речей, штучний інтелект, когнітивні обчислення, аналітика великих даних мають потенціал для зміни парадигми в авіаційній галузі, створюючи нові механізми, щоб зробити її не тільки ефективнішою, але й безпечнішою[2]. Проблеми зворотної сумісності є однією з головних проблем, які враховуються при розробці сучасних інформаційних систем, зазвичай вимагають використання стандартних інтерфейсів для забезпечення бездоганної інтеграції. Щоб вирішити цю проблему, слід використовувати стандартизовані формати формат обміну даними [3]. У мережевих системах, таких як авіація, стандартні формати даних, прийняті в усій галузі, також є вирішальними. Формат SSIM (Standard Schedule Information Manual, скорочено

SSIM) Міжнародної асоціації повітряного транспорту (International Air Transport Association, скорочено IATA) є одним із таких форматів, який дозволяє авіакомпаніям ефективно отримувати доступ до глобальних даних розкладу та використовувати їх.

IATA SSIM– це стандартний формат, який використовується авіакомпаніями для обміну розкладами польотів із Глобальними системами розподілу (англ. Global distribution system, скорочено GDS) та іншими авіаційними системами, що дозволяє створювати маршрути та виконувати інші функції, пов'язані з розкладом польотів. Формат SSIM включає структуровані елементи даних, такі як номер рейсу, час відправлення та прибуття, код аеропорту, тип літака та іншу інформацію[4]. Ці елементи визначені в певному порядку, утворюючи структурну ієрархію. Формат SSIM є гнучким і може враховувати зміни в розкладі рейсів залежно від частоти рейсів, робочих днів, обладнання літака та сезонних змін.

Формат SSIM також використовується для оновлення систем управління аеропортом і повітряним рухом актуальною інформацією про польоти. Крім того, авіакомпанії використовують формат SSIM для спільного використання кодів і інтерлайн-угод з іншими авіакомпаніями. Таким чином SSIM є ключовим механізмом зв'язку та обміну даними для авіакомпаній.

Незважаючи на свої переваги, формат SSIM також створює деякі складнощі в управлінні даними розкладів для авіакомпаній. Однією з найбільших проблем є високий рівень досвіду, необхідний для розробки та керування даними SSIM. Оскільки, це текстовий файл із фіксованою позицією (без роздільників), із різними типами записів, він потребує спеціального інструментарію перетворення даних, що ускладнює їх належне введення та подальший аналіз. Крім того, у SSIM можуть існувати певні рівні збігу та повторення даних, які потребують дедуплікації для використання в аналітичних системах. Це також може призвести до збільшення витрат на навчання персоналу для авіакомпаній.

Постановка проблеми

Цифрова трансформація авіаційної галузі та швидке поширення веб-додатків і онлайн-послуг створюють значні проблеми для використання традиційного формату SSIM-файлів для збереження даних розкладу польотів.

Основна проблема полягає в складній текстовій структурі SSIM із фіксованою шириною, яка, незважаючи на точність, за своєю суттю не сумісна з сучасними веб-технологіями та форматами обміну даними. Ця несумісність перешкоджає ефективній інтеграції, обробці та обміну даними в між цифровими авіаційними платформами. Відсутність сумісності між SSIM і такими зручними для Інтернету форматами, як JSON, створює складнощі в розробці таких нових програм, як оновлення розкладу в реальному часі, розширена аналітика авіа польотів та програм персоналізованих послуг для пасажирів.

Крім того, очевидна все більша залежність авіаційної промисловості від прийняття рішень на основі аналітики даних і потреба в гнучкому реагуванні на непередбачені ситуації[5]. Ця проблема також вимагає більш гнучкого та доступного формату для інформації про розклад.

Одним з рішень цієї проблеми є обмін інформацією у стандартизованому форматі JSON, оскільки JSON незамінним інструментом для різноманітних обмінів даними [6].

Цілі та проблеми дослідження

Це дослідження ставить перед собою наступні цілі:

1. Розробити надійну методологію для перетворення файлів SSIM у формат, який відповідає сучасним веб-стандартам, зокрема JSON;
2. Забезпечити збереження цілісності та точності вихідних даних графіку авіа польотів при їх перетворенні з SSIM формату у формат JSON;
3. Розробити програмне рішення-систему перевірки концепції, яка буде працювати за розробленою методологією та підтвердити її відповідність першим двом пунктам.

Виклад основного матеріалу

Для перетворення SSIM-файлу у формат JSON, виділимо чотири основних етапи процесу: перевірка вхідних даних; вичитування даних з рядків SSIM-файлу; аналіз та групування вчитаних даних; запис вчитаних даних у форматі JSON. Цей процес відображено на Рис. 1.

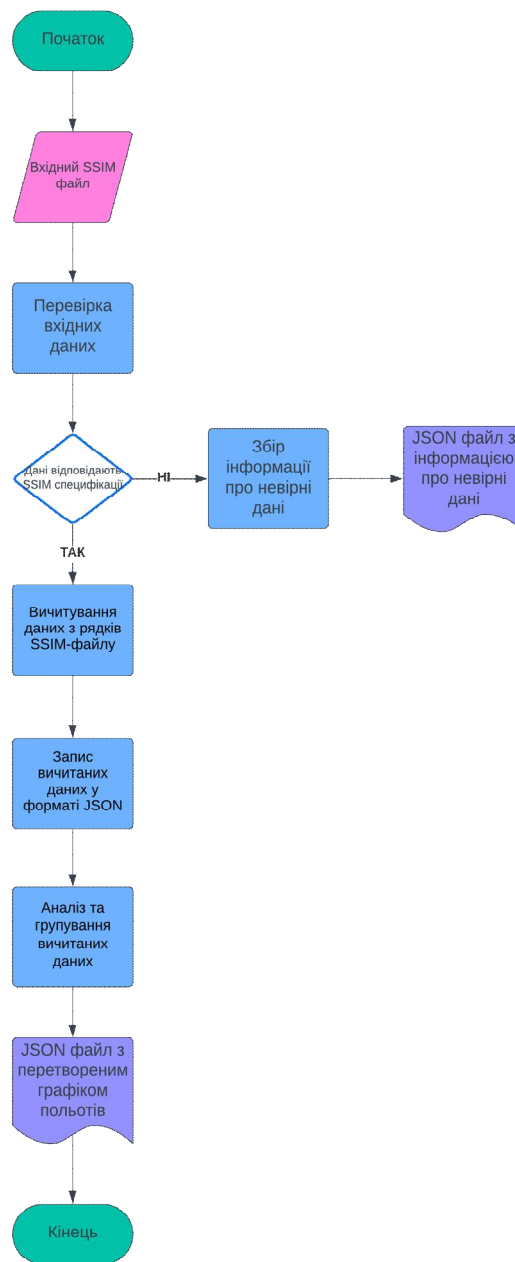


Рис. 1. Блок-схема процесу перетворення SSIM файлу у формат JSON

Для розробки системи перевірки концепції PoC будемо використовувати мову програмування Java. Причина такого вибору полягає в тому, що найвидатніша особливість Java полягає в тому, що вона не залежить від платформи, тобто має властивості «Напиши один раз– запускай де завгодно» (англ. Write once, run anywhere, скорочено WORA) [7].

Перевірка вхідних даних

Перший етап полягає у перевірці вхідних даних SSIM-файлу. Перевірка даних важлива як для прийняття рішень в процесі роботи програми, так і для перевірки самого процесу розробки програмного забезпечення[8].

Перевірка даних включає в себе перевірку довжини кожного рядка SSIM-файлу, оскільки, згідно специфікації, вона має бути рівно 200 символів. Наступним етапом є перевірка наявності всіх обов'язкових блоків SSIM-файлу. У SSIM-файлі використовується п'ять типів записів. Це:

Запис заголовка, запис перевізника, запис етапу польоту, запис даних сегмента та запис трейлера: 1-й блок– запис заголовка– тип запису 1 (обов’язковий); 2-й блок– запис оператора– тип запису 2 (обов’язковий); 3-й і– наступні блоки – запис етапу польоту; (а) запис про відрізок польоту– тип запису 3 (обов’язковий); (b) Запис даних сегмента– тип запису 4; наступний блок– трейлер-запис– тип запису 5 (обов’язковий)

Для перевірки кожного запису розробимо класи перевірки. Кожен клас буде перевіряти лише один тип запису для того, щоб відповідати принципу єдиної відповідальності(англ. Single Responsibility Principle). Цей принцип звучить як «Клас повинен мати лише одну причину для змін» [9].

Вичитування даних з рядків SSIM-файлу

Для вичитування даних з рядків SSIM-файлу будемо проводити їх синтаксичний аналіз згідно позиційних параметрів елементів визначених у SSIM-специфікації. Для цього сформуємо набір методів аналізу і представимо їх Таблиці 1.

Таблиця 1

Аналіз Синтаксису Рядка SSIM-файлу

Назва Методу	Опис	Індекси Підрядка	Примітки
recordType	Отримує тип запису	0-1	
origin	Отримує код аеропорту відправлення	36-39	
operatingAirline	Отримує код авіаліній, що експлуатують рейс	2-4	
destination	Отримує код аеропорту призначення	54-57	
flightNumber	Отримує повний номер рейсу	2-4, 5-9	Конкатенує код авіаліній та номер рейсу, видаляє пробіли у номері
onwardFlightNumber	Отримує номер наступного рейсу, якщо є	137-139, 140-144	Конкатенує код авіаліній та номер рейсу, повертає null, якщо пусто
fleet	Отримує інформацію про флот	72-75	
dayNrs	Отримує номери днів експлуатації	28-35	
departureTime	Отримує запланований час вильоту	43-47	
departureOffset	Отримує зміщення часового поясу вильоту	48-52	
departureOffsetSign	Отримує знак зміщення часового поясу вильоту	47-48	"+" або "-"
arrivalTime	Отримує запланований час прильоту	61-65	
arrivalOffset	Отримує зміщення часового поясу прильоту	66-70	
arrivalOffsetSign	Отримує знак зміщення часового поясу прильоту	65-66	"+" або "-"
fromDate	Отримує дату початку експлуатації рейсу	14-21	Видаляє пробіли на початку та в кінці
toDate	Отримує дату завершення експлуатації рейсу	21-28	Видаляє пробіли на початку та в кінці
serviceType	Отримує тип сервісу	13-14	Видаляє пробіли на початку та в кінці
aircraftConfig	Отримує конфігурацію літака	172-192	
aircraftOwner	Отримує код власника літака	128-130	
getLine	Повертає оригінальний рядок з файлу SSIM	Весь рядок	Корисно для налагодження або логування

Аналіз та групування вчитаних даних

Після того, як ми провели синтаксичний аналіз рядків SSIM-файлу, ми маємо провести групування отриманих даних. Спочатку з'єднаємо парні польоти на основі даних отриманих методом `onwardFlightNumber` на етапі синтаксичного аналізу. Для цього використаємо алгоритм описаний блок-схемою, зображеною на Рис. 2

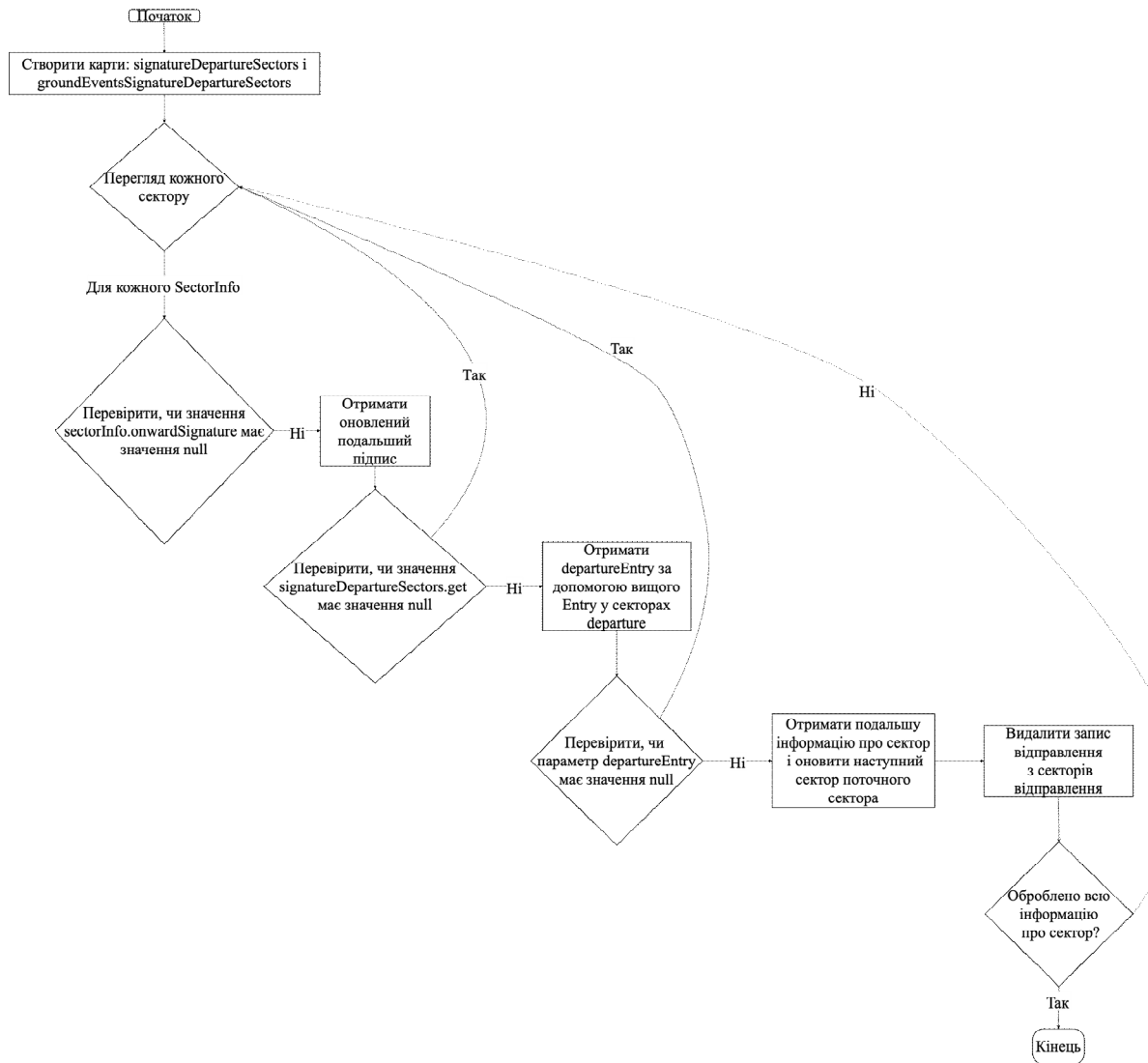


Рис.2. Блок-схема алгоритму з'єднання парних польотів

Наступним кроком поєднаємо пари польотів у лінію польотів(англ. flying line), тобто у послідовність польотів, яку здійснює певний авіа борт протягом тижня. Далі проведемо з'єднання неповних ліній польотів, тобто ліній польотів, що працюють не кожного дня тижня. Для цього використаємо Метод гілок і меж (англ. Branch-and-Bound). Метод гілок та меж є одним із найбільш часто використовуваних алгоритмів для вирішення NP-складних задач оптимізації [10].

Запис вчитаних даних у форматі JSON

Для запису вчитаних та згрупованих даних використовуємо бібліотеку мови Java, а саме Jackson-core.

Для відображення класів використаємо класову діаграму уніфікованої мови моделювання (англ. Unified Modeling Language, скорочено UML), тому що вона стала де-факто стандартом репрезентації діаграм програмної інженерії [11]. Діаграма класів програмного рішення на мові програмування Java, що реалізує кроки описані вище зображена на Рис.3

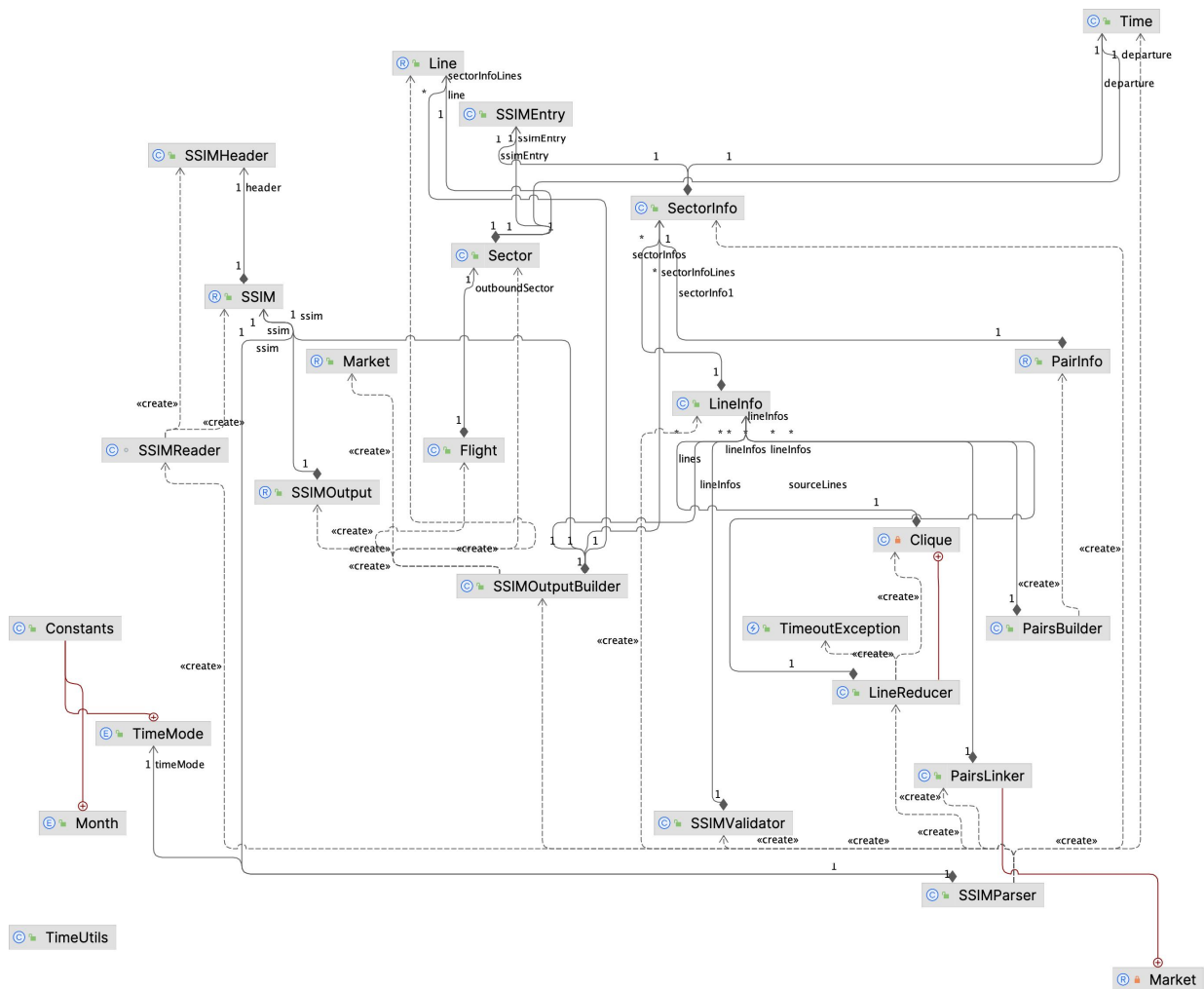


Рис.3. Діаграма класів розробленого програмного рішення

Результати та обговорення

Для перевірки розробленої програмної системи використаємо наступні вхідні дані SSIM-файлу, зображені на Рис.4.

В результаті вичитування, синтаксичного аналізу та групування даних вхідного файлу було знайдено 5 ліній польотів, які разом містять 41 політ. Знайдені лінії польотів:

Id: 5, Базовий аеропорт: SPX, Номер борту: QFC

[4, 7]

d4 16:35 FCO->WAW d4 19:05 C41444 тривалість: 150 хв, № лінії: 5, № наступного рейсу: C41443

d4 19:40 WAW->FCO d4 22:00 C41443 тривалість: 140 хв, № лінії: 5, № наступного рейсу: C48523

d7 10:50 SPX->FCO d7 13:20 C49311 тривалість: 210 хв, № лінії: 5, № наступного рейсу: C42816

Id: 3, Базовий аеропорт: KTN, Номер борту: 0IA

[1, 2, 3, 4, 5, 6, 7]

d1 14:40 KTN->IAS d1 18:00 C43701 тривалість: 140 хв, № лінії : 3, № наступного рейсу: C43697

d2 20:55 KTN->IAS d3 00:15 C43701 тривалість: 140 хв, № лінії: 3, № наступного рейсу: C43651

d3 21:05 KTN->IAS d4 00:25 C43701 тривалість: 140 хв, № лінії: 3, № наступного рейсу: C43661

d4 22:10 KTN->IAS d5 01:30 C43701 тривалість: 140 хв, № лінії: 3, № наступного рейсу: C43651

d5 14:40 KTN->IAS d5 18:00 C43701 тривалість: 140 хв, № лінії: 3, № наступного рейсу: C43697

d6 20:55 KTN->IAS d7 00:15 C43701 тривалість: 140 хв, № лінії: 3, № наступного рейсу: C43661

Від авіаційних до веб стандартів: Трансформація SSIM-файлів в JSON формат

d7 21:05 KTN->IAS d1 00:25 C43701 тривалість: 140 хв, № лінії: 3, № наступного рейсу: C43663
Id: 4, Базовий аеропорт: IAS, Номер борту: OIA

[4, 7]

d4 13:55 IAS->IST d4 16:45 C43717 тривалість: 110 хв, № лінії: 4, № наступного рейсу: C41837
d4 17:20 IST->IAS d4 17:55 C41837 тривалість: 95 хв, № лінії: 4, № наступного рейсу: C43685
d7 14:25 IAS->IST d7 17:15 C43717 тривалість: 110 хв, № лінії: 4, № наступного рейсу: C41837
d7 17:50 IST->IAS d7 18:25 C41837 тривалість: 95 хв, № лінії: 4, № наступного рейсу: C43709
Id: 2, Базовий аеропорт: CRA, Номер борту: 6CR

[1, 3, 4, 5, 7]

d1 12:10 CRA->KTN d1 13:20 C43751 тривалість: 130 хв, № лінії: 2, № наступного рейсу: C41189
d1 13:55 KTN->CRA d1 16:50 C41189 тривалість: 115 хв, № лінії: 2, № наступного рейсу: C48844
d1 18:00 CRA->LTN d1 19:15 C48844 тривалість: 195 хв, № лінії: 2, № наступного рейсу: C43754
d3 12:10 CRA->KTN d3 13:20 C43751 тривалість: 130 хв, № лінії: 2, № наступного рейсу: C41189
d3 13:55 KTN->CRA d3 16:55 C41189 тривалість: 120 хв, № лінії: 2, № наступного рейсу: C48844
d3 18:00 CRA->LTN d3 19:15 C48844 тривалість: 195 хв, № лінії: 2, № наступного рейсу: C43754
d4 12:30 CRA->KTN d4 13:40 C43751 тривалість: 130 хв, № лінії: 2, № наступного рейсу: C41189
d4 14:15 KTN->CRA d4 17:10 C41189 тривалість: 115 хв, № лінії: 2, № наступного рейсу: C48844
d4 18:00 CRA->LTN d4 19:15 C48844 тривалість: 195 хв, № лінії: 2, № наступного рейсу: C43754
d5 12:10 CRA->KTN d5 13:20 C43751 тривалість: 130 хв, № лінії: 2, № наступного рейсу: C41189
d5 13:55 KTN->CRA d5 16:50 C41189 тривалість: 115 хв, № лінії: 2, № наступного рейсу: C48844
d5 18:00 CRA->LTN d5 19:15 C48844 тривалість: 195 хв, № лінії: 2, № наступного рейсу: C43754
d7 12:30 CRA->KTN d7 13:40 C43751 тривалість: 130 хв, № лінії: 2, № наступного рейсу: C41189
d7 14:15 KTN->CRA d7 17:10 C41189 тривалість: 115 хв, № лінії: 2, № наступного рейсу: C48844
d7 18:00 CRA->LTN d7 19:15 C48844 тривалість: 195 хв, № лінії: 2, № наступного рейсу: C43754
Id: 1, Базовий аеропорт: PRG, Номер борту: QNA

[2, 4, 6, 7]

d2 18:50 PRG->BUD d2 20:35 C42360 тривалість: 105 хв, № лінії: 1, № наступного рейсу: C42359
d2 21:10 BUD->PRG d2 22:55 C42359 тривалість: 105 хв, № лінії: 1, № наступного рейсу: C48423
d4 06:00 PRG->WAW d4 08:20 C41458 тривалість: 140 хв, № лінії: 1, № наступного рейсу: C41457
d4 08:55 WAW->PRG d4 11:25 C41457 тривалість: 150 хв, № лінії: 1, № наступного рейсу: C41050
d4 12:55 PRG->KTW d4 15:00 C41050 тривалість: 125 хв, № лінії: 1, № наступного рейсу: C41049
d4 15:35 KTW->PRG d4 17:40 C41049 тривалість: 125 хв, № лінії: 1, № наступного рейсу: C43157
d6 18:50 PRG->BUD d6 20:35 C42360 тривалість: 105 хв, № лінії: 1, № наступного рейсу: C42359
d6 21:10 BUD->PRG d6 22:55 C42359 тривалість: 105 хв, № лінії: 1, № наступного рейсу: C41458
d7 06:00 PRG->WAW d7 08:20 C41458 тривалість: 140 хв, № лінії: 1, № наступного рейсу: C41457
d7 08:55 WAW->PRG d7 11:25 C41457 тривалість: 150 хв, № лінії: 1, № наступного рейсу: C41050
d7 12:55 PRG->KTW d7 15:00 C41050 тривалість: 125 хв, № лінії: 1, № наступного рейсу: C41049
d7 15:35 KTW->PRG d7 17:40 C41049 тривалість: 125 хв, № лінії: 1, № наступного рейсу: C43157

Кількість ліній польотів: 5

Загальна кількість польотів: 41

Кількість польотів по днях: {ПОНЕДІЛОК=4, ВІВТОРОК=3, СЕРЕДА=4, ЧЕТВЕР=12, П'ЯТНИЦЯ=4, СУБОТА=3, НЕДІЛЯ=11}

В результаті запису вчитаних даних в формат JSON, отримуємо наступну JSON-стрічку:

```
{"markets":[{"code1":"BUD","code2":"PRG","flights":[{"flightId":0},{"flightId":3}],"code1":"FCO","code2":"SPX","flights":[{"flightId":26}],"code1":"FCO","code2":"WAW","flights":[{"flightId":25}],"code1":"CRA","code2":"KTN","flights":[{"flightId":6},{"flightId":8},{"flightId":10},{"flightId":12},{"flightId":14}],"code1":"CRA","code2":"LTN","flights":[{"flightId":7},{"flightId":9},{"flightId":11},{"flightId":13},{"flightId":15}],"code1":"PRG","code2":"WAW","flights":[{"flightId":1},{"flightId":2}
```


- [4] International Air Transport Association, Standard schedules information manual, Ref. No. 9179-21, Montreal – Geneva, 2011. ISBN: 978-92-9233-496-3.
- [5] B. Ervural, "Prediction of Airline Flow Using Data Analytics Methods," 2023. [Online]. Available: 10.1007/978-3-031-43940-7_16.
- [6] T. Bray, "The javascript object notation (json) data interchange format," No. rfc7159, 2014. <https://doi.org/10.17487/rfc7159>
- [7] S. Khoirom, M. Sonia, B. Laikhuram, J. Laishram, and T.D. Singh, "Comparative analysis of Python and Java for beginners," *Int. Res. J. Eng. Technol*, vol. 7, no. 8, pp. 4384-4407, 2020.
- [8] M. Patil and N. Yogi, "Importance of Data Collection and Validation for Systematic Software Development Process," *Int. J. Comput. Sci. & Inf. Technol.*, vol. 3, 2011. [Online]. <https://doi.org/10.5121/ijcsit.2011.3220>
- [9] R.C. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, Boston, 2018.
- [10] J. Clausen, "Branch and Bound Algorithms—Principles and Examples," Univ. Copenhagen, 1999.
- [11] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, Reading, Mass, Addison Wesley Longman Inc., 1999.

Oleksii Dzhus¹, Mykhailo Lobur²

¹ Computer Aided Design Department, Lviv Polytechnic National University, Ukraine, Lviv, S. Bandery street 12, E-mail: oleksii.p.dzhus@lpnu.ua

² Computer Aided Design Department, Lviv Polytechnic National University, Ukraine, Lviv, S. Bandery street 12, E-mail: mykhalo.v.lobur@lpnu.ua, ORCID 0000-0001-7516-1093

FROM AVIATION TO WEB STANDARDS: TRANSFORMATION OF SSIM FILES INTO JSON FORMAT

Received: March 01, 2024 / Reviewed: April 01, 2024 / Accepted: April 05, 2024

© *Dzhus O., Lobur M., 2024*

Abstract. Considering dynamic changes in the aviation sector, successful data exchange and integration planning plays a pivotal role in maintaining operational efficiency of airlines. Although the SSIM (Standard Schedules Information Manual) file is considered a reliable format, it has certain limitations regarding interaction and compatibility with other web technologies and programs due to its text format strictly defined by the specification. In this work, we propose an approach to convert SSIM files into the JSON (JavaScript Object Notation) format.

One of the central aspects of this work is a detailed transformation model that considers all the important structures and levels of information of the SSIM file, conducts their syntactic analysis, and converts them into a flexible and easily recoverable JSON format.

To transform a set of aviation flights into a combined flying line, a modified branch and bound method is used.

With a series of examples and experiments, the article aims to demonstrate the effectiveness of this methodology in terms of preserving data integrity and ensuring their accuracy. In particular, the problems faced during the transformation, for example, how to consider the consistency of data rows with fixed width in the SSIM file, are considered.

The transition from aviation-oriented formats, such as the SSIM file, to global web formats, such as JSON, is a process that goes beyond technologies and aims to increase the flexibility and convenience of collaboration, as well as data processing in the aviation industry. This research helps to implement this change, offering a simple, effective, and scalable approach for converting airline schedules into a format that by its nature meets the needs of modern information applications and systems.

Key words: SSIM, JSON, airlines, data format, syntactic analysis, branch and bound method