

АО «СПЕЦИАЛЬНЫЕ СИСТЕМЫ»

Парсер тарифных файлов EWSD

Техническая архитектура

Оглавление

Ограниченное использование ПО	2
Описание технической архитектуры.....	2
Среда использования.....	2
Место EWSD-парсера в архитектуре входной подсистемы Антифрод-системы	3

Ограниченное использование ПО

Настоящий документ является результатом интеллектуальной деятельности, исключительное право на которое принадлежит акционерному обществу «Специальные системы» (именуемое далее правообладатель).

Любое использование (как полностью, так и в части) настоящего документа (в частности: копирование, воспроизведение, распространение, доведение до всеобщего сведения и т.д., в цифровой форме и/или на бумажных носителях) допускается только по соглашению с правообладателем. Нарушение исключительного права преследуется в соответствии с законодательством Российской Федерации, нормами международного права.

Правообладатель вправе вносить изменения в Программное Обеспечение (далее – ПО), настоящую документацию без предварительного уведомления Лицензиата.

Описание технической архитектуры

Среда использования

Парсер EWSD создавался для использования в Антифрод-системах и является одним из небольших компонентов таких систем. Обсуждаемый класс Антифрод-систем использует постобработку биллинговых данных и не относится к онлайн системам. Прежде чем входные данные попадут в Антифрод-систему они должны быть собраны в пакеты данных за некоторый интервал времени, после чего поставлены во входную подсистему, разобраны, загружены в БД и стать доступными для аналитической обработки с целью выявления мошеннических действий на сетях оператора связи. При разработке входной подсистемы было необходимо решать следующие вопросы:

1. в качестве минимальной единицы входных данных рассматривать файл с порцией биллинговых данных;
2. изолировать друг от друга следующие процессы конвейера обработки файлов входных данных, для этого необходимо реализовать их как атомарные сервисы, не взаимодействующие напрямую друг с другом:

- 2.1. сервис обнаружения новых файлов входных данных и их классификацию на принадлежность к тому или иному типу данных;
- 2.2. сервис доставки файлов входных данных для их обработки;
- 2.3. сервис разбора (парсинга) входных файлов данных к csv-формату, удобному для пакетной загрузки в БД;
- 2.4. сервис загрузки csv-файлов в БД.

Следует иметь ввиду, что входная подсистема Антифрод-системы должна:

- во-первых, уметь обрабатывать большое разнообразие типов файлов входных данных;
- во-вторых, легко расширяться без внесения изменений в код ядра Антифрод-системы при появлении новых типов данных;
- иметь возможность вертикального и горизонтального масштабирования.

Место EWSD-парсера в архитектуре входной подсистемы Антифрод-системы

EWSD-парсер, как парсеры любого другого типа во входной подсистеме, должен удовлетворять следующим архитектурным требованиям:

1. Иметь способность динамического включения в состав системы без внесения изменений в код сервиса разбора (парсинга) входных файлов данных (Late Binding). Для реализации этого требования все парсеры:
 - 1.1. Наследуются от единого интерфейса `IParser`, в котором определен единственный метод `ParseStatistics Parse()`;
 - 1.2. Экземпляр парсера загружается на этапе выполнения кода сервиса разбора (парсинга) входных файлов данных при обнаружении файла, готового к парсингу с помощью метода `Assembly dll = Assembly.LoadFile(pluginDllFN)`, где `pluginDllFN` – имя DLL, зарегистрированной для парсинга входных файлов типа тарифный файл EWSD.
 - 1.3. DLL-библиотеки парсеров хранятся в едином каталоге парсеров входной подсистемы. В процессе развития Антифрод-системы, при необходимости добавить новый тип входных данных, создается DLL

парсера для нового типа данных. Эта новая DLL-парсера добавляется в единый каталог парсеров и регистрируется во входной подсистеме для разбора файлов нового типа. Таким образом, отпадает необходимость изменять код сервиса разбора (парсинга) входных файлов.

2. Выполнять единицу работы (Unit of Work) – разбор данных одного входного файла одного типа (Single Responsibility). Каждый тип парсера унаследован от абстрактного базового класса RootParser, конструктор которого имеет следующий синтаксис:

```
public RootParser(string inFN, string loadFN, string logFN, XElement cfgParams)
```

Парсер получает из вызывающего кода параметры для выполнения единицы работы – разбора входного файла и его сохранении в выходном csv-файле. Для конфигурирования каждого парсера используется файл конфигурации в xml-формате, специфичный для данного типа входных данных.

3. Быть слабосвязанным с остальными компонентами входной подсистемы. Сам парсер не взаимодействует с БД для регистрации своих действий (изолирован от БД) и взаимодействует только с файловой системой – читает входной файл и записывает выходной файл. Действия по регистрации результатов парсинга выполняются в общем вызывающем коде сервиса разбора (парсинга) входных файлов.
4. Использовать единый механизм регистрации диагностики. Это требование реализовано с помощью фреймворка NLog.