Обзор Kaitai Struct (технология разбора бинарных данных на основе декларативных спецификаций их форматов)

Михайлов А.А.

Kaitai Strcut -

Декларативный язык описания бинарных форматов.

Позволяет:

- Описать формат
- Проверить с помощью средств визуализации (ksv)
- Скомпилировать в библиотеку на целевом языке (ksc)
- Использовать полученный API

Лицензия

- Компилятор – GPLv3+
- Библиотеки для чтения файлов – MIT or Apache v2

Декларативное и императивное описание

Императивное

- Как прочитать или записать формат
- Привязано к конкретному языку
- Как правило, реализуется вручную

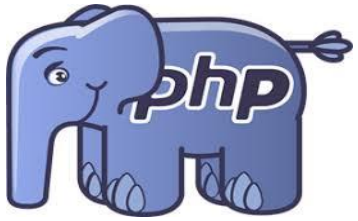
Декларативное

- Что хранится в структуре
- Не привязано к конкретному языку
- Нужен какой-то способ претворять в жизнь:
  - интерпретатор
  - компилятор

Для чего предназначен Kaitai Struct

- Коммуникационные протоколы
- Упакованные embedded firmware
- Контейнерные форматы файлов
- Исполняемые файлы (executables, objects, байт-код, ...)
- Файлы контента (базы данных, таблицы, тексты, графика, текстуры, ...)
- Файловые системы

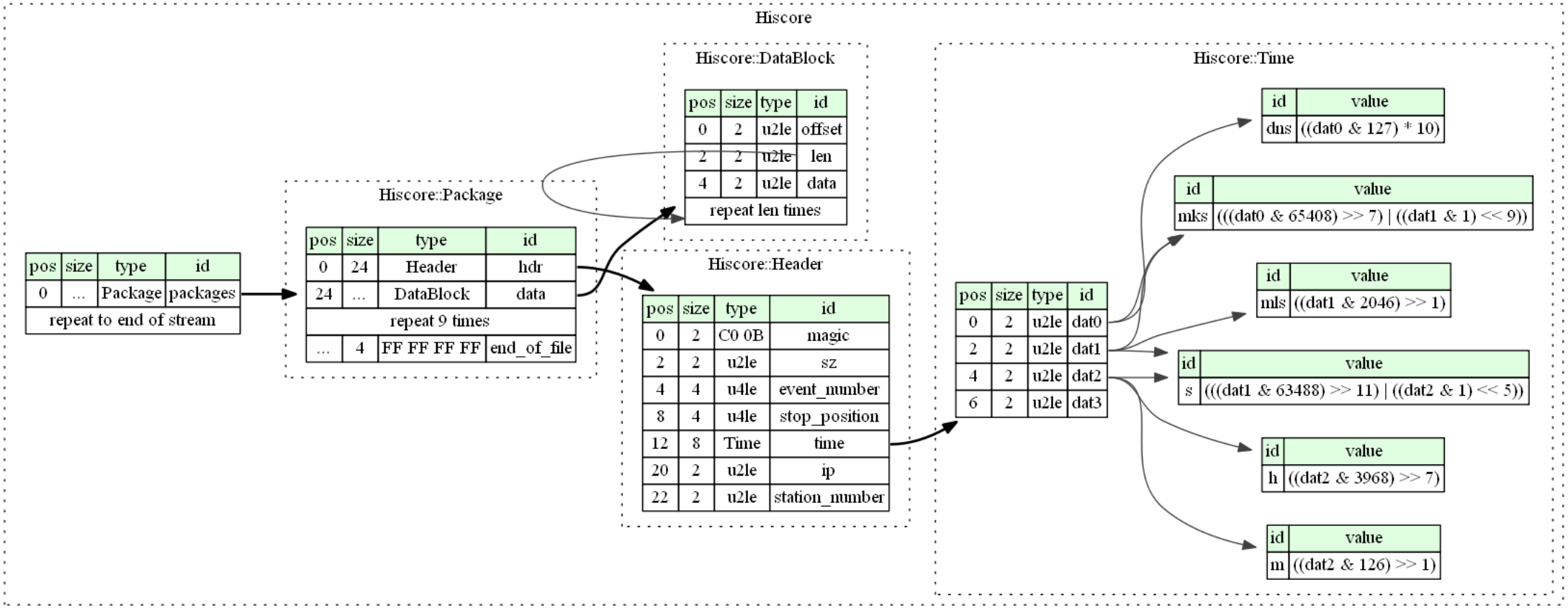
Целевые языки



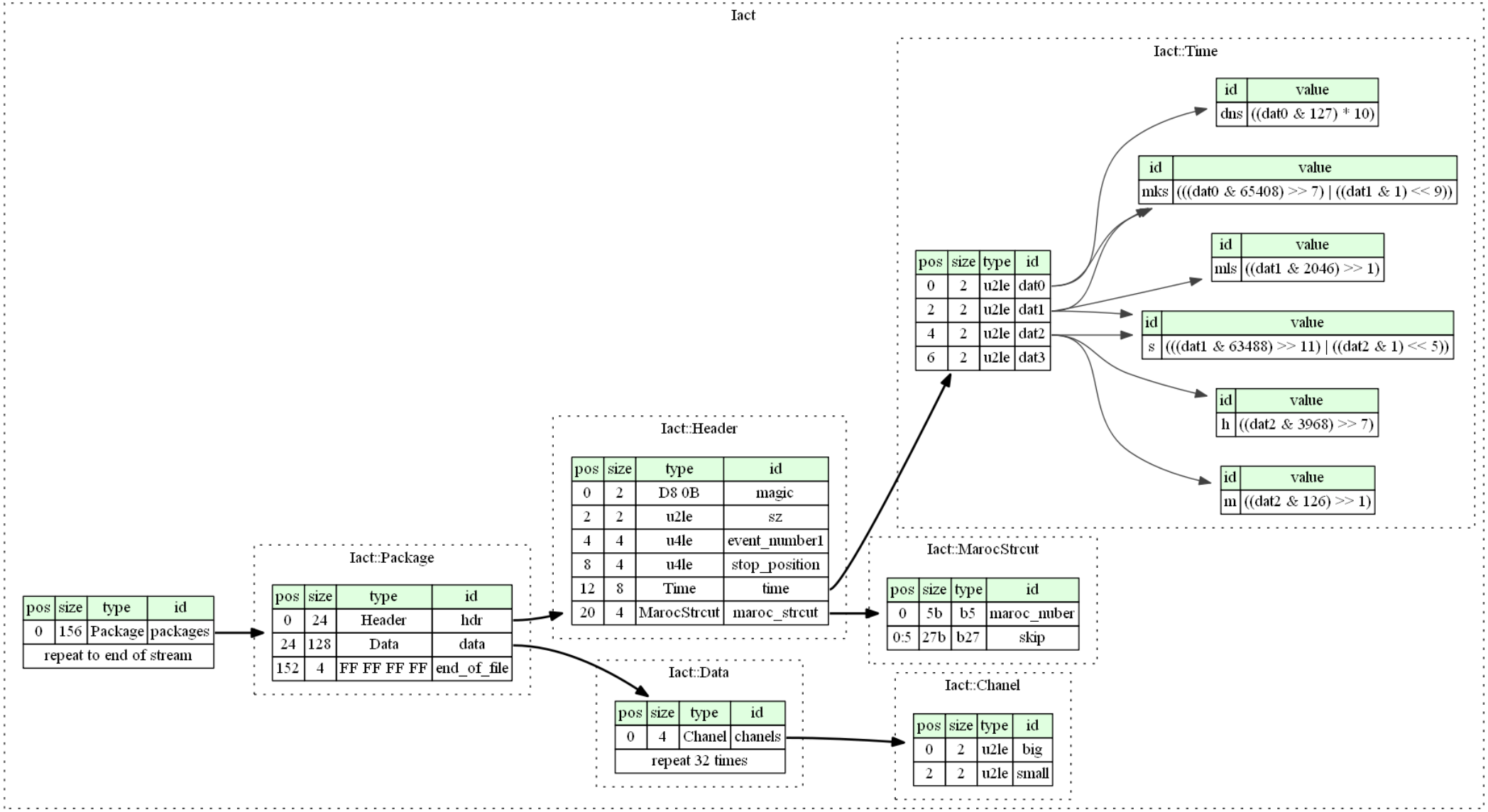
Ближайшие аналоги

- Проприетарные hex-редакторы: 010 Editor, Synalysis, Hexinator
- Ряд библиотек для одиночных языков: Preon, jBinary, и т. д.
- Стандарт DFDL (Data Format Description Language).разрабатывается для описания текстовых и бинарных данных, используемых в GRID-системах. Имеется коммерческая реализация парсера DFDL в составе продукта IBM Integration Bus, а также открытая реализация.
- MFL (Message Format Language) в составе продукта WebLogic Integration фирмы Oracle.
- Декларативный язык FlexT – инструмент анализа и документирования бинарных форматов данных

Формат бинарных данных HiSCORE



Формат бинарных данных IACT



Пример программы чтения файла HiSCORE

```
public static void main(String[] args) throws IOException {
    Hiscore hiscore = Hiscore.fromFile(FILE_NAME);
    int pckgNumber = 0;
    for (Hiscore.Package pckg : hiscore.packages()) {
        System.out.println("Package: " + pckgNumber);
        System.out.println(String.format("Magic: %02X %02X", pckg.hdr().magic()[0], pckg.hdr().magic()[1]));
        System.out.println("Size: " + pckg.hdr().sz());
        System.out.println("Event number: " + pckg.hdr().eventNumber());
        System.out.println("Stop position: " + pckg.hdr().stopPosition());
        System.out.println(String.format("H %d: M %d: S %d: DNS %d: MKS %d: MLS %d",
                pckg.hdr().time().h(), pckg.hdr().time().m(),
                pckg.hdr().time().s(), pckg.hdr().time().dns(),
                pckg.hdr().time().mks(), pckg.hdr().time().mls()));
        System.out.println("IP: " + pckg.hdr().ip());
        System.out.println("Station number: " + pckg.hdr().stationNumber());
        int size = pckg.data().size();
        int dataBlockNumber = 0;
        for (Hiscore.DataBlock data: pckg.data()) {
            System.out.println("Data block: " + dataBlockNumber);
            System.out.println("Offset: " + data.offset());
            System.out.println("Length: " + data.len());
            int n = data.len() > DATA_COUNT_TO_VIEW ? DATA_COUNT_TO_VIEW : size;
            System.out.print("[");
            for (int i = 0; i < n; i++) {
                System.out.print(data.data().get(i) + ", ");
            }
            System.out.print("...]");
            System.out.println();
            ++dataBlockNumber;
        }
        ++pckgNumber;
    }
```