

Упрощение работы с конфигурационными файлами в программных комплексах

Лукьянец Е.А.

(Университет ИТМО, г. Санкт-Петербург)

Научный руководитель – д.т.н., Матвеев Ю.Н.

(Университет ИТМО, г. Санкт-Петербург)

Введение

Исследования выполнены за счет стартового финансирования университета ИТМО в рамках НИР № 618278 «Синтез эмоциональной речи на основе генеративных состязательных сетей». Целью исследования является поиск удобного способа работы с файлами конфигурации в программных комплексах.

Базовые положения исследования

В работе любого программного комплекса существует компонент, отвечающий за чтение и валидацию конфигурационных файлов. Обычно написание этого компонента является сложной в реализации задачей, а в итоговом коде часто встречаются ошибки и опечатки, которые проявляются в виде ошибок в функционировании всего комплекса.

Компонент, отвечающий за работу с конфигурационными файлами, должен выполнять несколько основных функций:

- валидировать «схему» (англ. Schema) документа — проверять, что необходимые поля конфигурации заполнены данными;
- валидировать значения — проверять, что значения, указанные в конфигурационном файле, соответствуют ожиданиям (например, что некоторое число не менее 0).

Дополнительным полезным качеством конфигурационных файлов является лёгкость их чтения разработчиком.

Существует большое количество различных форматов, которые можно использовать для конфигурационных файлов:

- XML [1],
- JSON [2],
- YAML [3],
- прочие.

Валидация «схемы» — наиболее трудоёмкий и подверженный ошибкам процесс обработки конфигурационного файла, поэтому было бы плюсом, чтобы был стандарт, описывающий «схему» и валидирующий её.

XML поддерживает «схемы», например DTD [4], но, к сожалению, они очень неудобны в использовании и написании. YAML и JSON схемы не поддерживают, однако для JSON есть черновик стандарта, что значит, что он не стабилен и может измениться в будущем, что, опять же, не подходит. При этом наиболее удобным для чтения и редактирования разработчиками является формат YAML.

В качестве решения проблемы предлагается использовать текстовую сериализацию ProtoBuf [5]. Синтаксис текстовой сериализации ProtoBuf очень похож на JSON, а потому достаточно привычен и удобен. Валидации значений в этом стандарте нет, однако схема и её валидация — есть. Это связано с тем, что основное назначение ProtoBuf — это сериализация в бинарный вид сообщений клиент-серверного взаимодействия.

Выводы

В рамках проведённой работы были рассмотрены различные способы описания конфигурационных файлов, выявлены их достоинства и недостатки, а также предложен новый более удобный в использовании формат конфигурационных файлов.

Литература

1. XML.com [Электронный ресурс]. – Режим доступа: <https://www.xml.com>, своб.
2. Introducing JSON [Электронный ресурс]. – Режим доступа: <https://json.org>, своб.

3. The Official YAML Web Site [Электронный ресурс]. – Режим доступа: <https://yaml.org>, своб.
4. XML DTD [Электронный ресурс]. – Режим доступа: https://w3schools.com/xml/xml_dtd.asp, своб.