Joshua Moughler

Jacob Gnatz

CIS445 01

Tachun Lin

10/16/2018

Comparative Situational Analysis of JSON Alternatives

1. Learning objectives and expected outcome

The biggest thing we wanted to understand in this project was why you would use one type of data interchange language over the other. Going into this project it appeared as if there was these 4 different languages that all performed the same functions. Josh expected to come out of this project with the ability to say that for the most part YAML was the most superior language because of its readability.

1. Challenges encountered

As BSON and UBJSON are less popular serialization formats, finding working translators between these file types and JSON was a large hassle. In particular, BSON was one of the harder files to validate due to the format being a human unreadable format. This was overcome by encoding into BSON and decoding back from BSON into JSON, making sure that the encoding was done successfully.

1. Approaches taken to tackle the challenges

Our biggest challenge was we had originally made our topic statement too broad and too vague. Several times during the project we needed to reassess what the priorities were. It was not that we felt as we could not tackle various parts of the topic but because we, overtime, felt that different concepts we mentioned in the topic did not align with the overall intent of our research. We did not want the research to be mostly uniform and then a bunch of other things thrown in for the sake of maintaining every detail.

1. Lessons Learned

Taking more time to thoroughly study topics before coming up with a use case for them would benefit us more in the long run. As we initially set out to study configuration files, we set our sites on using JSON alternatives for this purpose, even though the applications for these extends well beyond the use case of system configuration. Taking a more broad study and narrowing down the focus after more general steps would be helpful in developing future research projects.

1. Responsibilities of each member

Josh created the original JSON file that both of us took and moved to our various languages that we studied. Josh also did the research into JSON, YAML, and XML. Jacob did the research into BSON and UBJSON. We both created our own powerpoint slides. We summarized our own research in this report. Each member was responsible for committing and pushing their own code deliverables, while Josh pushed the presentation and reports.

1. Summary of work and progress report

We learned the situational use cases for various languages.

JSON is a data interchange language. It is designed to be easily read but not edited. It was designed that way as the creator did not believe that comments were not useful for a data interchange language. It was designed by Douglas Crawford. Defined in RFC 7159. Based on JavaScript but is able to be used in many other languages now

YAML is a data interchange language but is not a substitute or the same as JSON with different syntax. YAML is designed to be used in configuration. An example of a config file that will be on many of your computers is \_config.yml and is used by Visual Studio to go out to GIT and download the bootstrap framework.

XML is a markup language which makes it different from JSON and YAML. And as a document markup language it is used by both Microsoft Office and by Open office. XML is more cryptic to read and does not support arrays and making it is not as user friendly when using it with other C family languages like C++ and Java.

UBJSON is a serialization derivative based on JSON that allows for more compacted (and thereby faster parsed) data serialization compatible fully with JSON. UBJSON in the average case compressed JSON to be 30% smaller, although this gain may be higher depending on the data types present in the original JSON. UBJSON achieves this compression by using marker-characters to specify data types beforehand, allowing parsers to know explicit information about the data it is translating instead of having to calculate from a text-representation.

BSON is a non-human readable binary serialization format intended to take JSON-like files and make them easily traversable and potentially more space efficient (if possible). BSON also adds new data types (date, byte-array) not found in the original JSON specifications, making it less compatible. It is the primary source of data transfer used by MongoDB, a large database service company currently serving companies such as Forbes and Comcast (among others). BSON translates data type, file length, and information about the value of each key-data pair into binary before storing such data, and it is through these additional binary markers that BSON achieves faster scannability.

1. References used in the project

Most of the research that we used was from the developer sources themselves. We also accessed a few comparison websites that would provide context for our project.

*JSON vs XML*. [Online]. Available: https://www.w3schools.com/js/js\_json\_xml.asp. [Accessed: 16-Oct-2018].

*Universal Binary JSON Specification*. [Online]. Available: http://ubjson.org/. [Accessed: 16-Oct-2018].

*BSON (Binary JSON): FAQ*. [Online]. Available: http://bsonspec.org/faq.html. [Accessed: 16-Oct-2018].

“BSON,” *BSON (Binary JSON) Serialization*. [Online]. Available: http://bsonspec.org/. [Accessed: 16-Oct-2018].

“Introducing JSON,” *JSON*. [Online]. Available: https://www.json.org/. [Accessed: 16-Oct-2018].

“JSON to YAML,” *YAML vs JSON*. [Online]. Available: https://www.json2yaml.com/yaml-vs-json. [Accessed: 16-Oct-2018].

“The Official YAML Web Site,” *The Official YAML Web Site*. [Online]. Available: http://yaml.org/. [Accessed: 16-Oct-2018].

1. What to do next: if you have finished what you promised to deliver, then what would you like to do next to extend/expand the project. If you have not finished, how would you plan to finish it in another two to three weeks.

If we were given another two or three weeks we could go into more detail surrounding the details of speed and efficiency. We would be able to provide more situational comparisons. We were able to give an overview of the intent of each language but we might be able to show how in a configuration file YAML is less rigid than than JSON. Or that when working with with databases BSON might be the more appropriate language.

1. Meeting schedule, agenda, and action plan

Discuss topic options 9/27/2018

Choose topic 10/1/2018

Project approval 10/2/2018

Refine Topic / Scope Meeting 10/2/2018

Start Research 10/2/2018

Start Powerpoint 10/2/2018

Define Key Talking Points 10/5/2018

Define Final Draft of Topic 10/11/2018

Start Code 10/10/2018

Start Report 10/15/2018

Final Planning Meeting 10/15/2018

Review Meeting 10/16/2018

Finish Powerpoint 10/16/2018

Finish Report 10/16/2018