Using JSON with NIEM

Abstract

The National Information Exchange Model (NIEM) currently uses XML (Extensible Markup Language) exclusively: for the NIEM reference model; Model Package Descriptions (MPDs) that define the content, structure, and semantics of information models and information exchanges; and information exchange package (IEP) instances. The NIEM Technical Architecture Committee (NTAC) is examining the use of an alternative data-interchange format, JavaScript Object Notation (JSON), for use with NIEM. The NTAC has developed an approach and roadmap for supporting the use of JSON with NIEM. In CY 2016, the NTAC plans to develop and publish initial guidance and a specification for NIEM-conformant JSON IEPs. The NTAC is also soliciting suggestions and feedback on using JSON with NIEM.

What is JSON?

JSON is a lightweight data-interchange format. It is programming language independent but uses conventions that are familiar to programmers. Like XML, JSON is a text format that is intended for use in machine-to-machine information exchanges. For some developers, JSON can be simpler than XML to learn, understand, develop, and deploy; but with the tradeoff that it has fewer capabilities.

The use of JSON has evolved from the similar technology mix that gave rise to information exchanges using XML. With XML, the dominance of web services and Service Oriented Architecture (SOA) solutions on premise provided the vehicles. Hence XML and NIEM also focused on building formal information exchanges within an information sharing enterprise.[1] For JSON, the commercial success of cloud-based web applications, Software as a Service (SaaS), and using RESTful services instead of SOAP web services has been the driver.

The use of JSON has increased dramatically since it became a format commonly used in web applications for exchanging data between the server and the client web browser. Many organizations are now routinely using JSON for their data-interchange format. The use of JSON is built directly into the web applications tools used, making the JSON transparent to developers. While this simplicity comes at the expense of capabilities, in many cases the additional capabilities of XML are not needed because the exchanges are mostly intra-organizational and not enterprise wide.

Some of the potential benefits of JSON are:

- It can lead to shorter code development and deployment cycle times compared to XML.
- It can lead to lower costs compared to XML.

[1] An information sharing enterprise is a group of organizations with business interactions that agree to exchange information, often using multiple types of information exchanges. The member organizations have similar business definitions for objects used in an information exchange and can agree on common definitions.

- JSON developers usually don't need to learn or understand XML or associated specifications such as XML Schema structures and datatypes, XSLT (Extensible Style Language), and XPath (XML Path Language).

Some of the limitations of JSON are:

- JSON has no finalized standard for specifying the permitted content, structure, and semantics of an information exchange – there is no standard for JSON equivalent to XML Schema.[2]
- Fewer structures and datatypes are available compared to XML.

Examining JSON and NIEM

Demand

NTAC is examining the use of JSON with NIEM due to the demand from the NIEM community:

- The use of JSON has become widespread and is increasing. If the NIEM program dismisses JSON, NIEM may lose some relevance and will be unable to support a growing segment of the NIEM community that prefers to use JSON for information exchanges.
- The NIEM PMO is receiving questions about how to create NIEM-conforming JSON, but as of today there is no such thing as "NIEM-conforming JSON".
- The NIEM program would like to support people who want to do what NIEM does, but using JSON instead of XML.

Challenges

Developing an approach to using JSON with NIEM presents the following challenges:

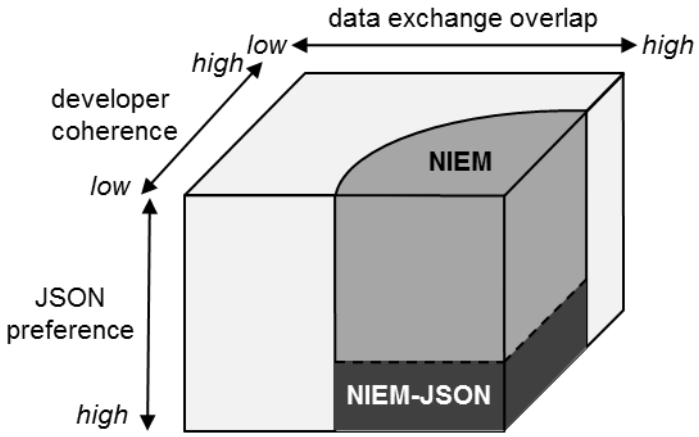
- The approach must not dilute the useful characteristics of JSON, negating the benefits of JSON's simplicity so much that the approach is rejected by JSON implementers.
- The approach must not dilute NIEM information sharing principles, negating the benefits of NIEM's support for interoperability so much that NIEM-conformance becomes meaningless.
- The approach must consider the fact that there is no accepted standard for a JSON equivalent to XML Schema

The case for using JSON and NIEM together

For any given information exchange, an organization or an information sharing enterprise could choose to use NIEM or not use NIEM. An organization could choose to use XML, JSON, or some other data-interchange format (although currently for NIEM-conformance it must be XML). When would an information sharing enterprise want to use NIEM and JSON together?

[2] JSON Schema (http://json-schema.org/) was first proposed as an Internet Engineering Task Force (IETF) draft standard in 2009, but the draft expired in 2013 without approval.

Attachment 1, *NIEM-JSON Tradespace*, analyzes and describes the NIEM-JSON tradespace. As illustrated in the figure, an organization may want to use NIEM when *data exchange overlap* has a tendency to be high and *developer coherence* has a tendency to be low; the organization may additionally want to use JSON (with NIEM) when the organization's a preference for JSON tends to be high.



Data exchange overlap has a tendency to be high when developers work on several exchanges that have concepts in common (whether all at once or over time). The harmonized component reuse provided by NIEM becomes important, because the effort of understanding a component used in one exchange need not be repeated when the same component is used in another exchange.

Developer cohesion tends to be low when the number of developers who must understand the exchanged data increases. Developer cohesion decreases further as the number of organizations involved rises, because sharing knowledge across organization boundaries is more difficult. The documentation framework provided by NIEM is valuable to developers who need rigorous specifications for data exchanges. The value also increases along with the length of time the exchange must be maintained, because developers will come and go, and because wise programmers write documentation for things they themselves must understand in the future.

JSON preference tends to be high when an organization already uses JSON-based information sharing architectures, uses JSON capabilities of development tools, has developers with JSON experience or preference, and/or uses JSON for non-NIEM-conformant information exchanges.

At this time, some organizations using NIEM for information sharing may want to use JSON as the data interchange format, but can't because they believe using NIEM is more important than using JSON. Alternatively, some organizations using JSON as the data interchange format may want to use NIEM, but can't because they believe using JSON is more important than using NIEM. Finally, some organizations are having to choose to use NIEM-conformant XML for some information exchanges and non NIEM-conformant JSON for others. Permitting NIEM-conformant JSON eliminates or lessens these compromises.

Finally, providing the ability to use JSON with NIEM can be viewed as consistent with a Unified Modeling Language (UML) model-driven approach whereby the data representation can be more independent of the data model, permitting the use of new data representation technologies as they become widely adopted.

Contract-first v. code-first interface development

The use of JSON raises a related issue: the concept of *contract*-first interface development versus *code*-first interface development. While XML and JSON can use either approach, JSON is more

popularly implemented using *code*-first interface development. However, using NIEM requires *contract*-first interface development. The benefits of NIEM and the interoperability use case for NIEM are only realized through contract-first interface development, regardless of whether XML or JSON is used.

The particular representation of NIEM does not obviate the need for stakeholders to ascertain, agree on, and document the content, structure, and semantics of an information exchange. Information Exchange Package Documentation (IEPD) or other Model Package Description (MPD) and the effort to develop it is required regardless of whether XML or JSON is being used.

With contract-first interface development, the IEPD is developed before any of the exchange partners develops an interface to send or receive the information exchange. The interfaces are then developed and implemented to exchange information in accordance with the IEPD.

With code-first interface development, one of the exchange partners develops an interface to support the information exchange. Once the interface development is complete, the documentation for the interface is generated (often automatically) to reflect the "as-built" interface. The representation of content (terminology, semantics, and structure) is usually consistent with the particular implementation of the interface and does not necessarily reflect any agreement with other exchange partners. The documentation is often limited to whatever the development tool generates, and frequently does not clearly define the semantics.

Although in theory it is possible to use code-first development and generate NIEM-conformant IEPD artifacts from the as-built interface, in practice this has not been practical to date.

Developers have a good reason to want to do code-first development because it is easier for them – they can do what they want. It is very developer-centric at the expense of the interoperability goals NIEM achieves through harmonized components and a documentation framework. Referring to the NIEM-JSON tradespace discussion, it is perfectly OK to use the code-first approach if you have high developer coherence and low data exchange overlap, where you have no need for NIEM. If you have decided to use NIEM, you likely must use the contract-first approach, regardless of whether you are using XML or (in the future) JSON.

Notwithstanding the above, the NTAC will continue to examine the potential for using NIEM with a code-first development approach by researching and evaluating options, soliciting suggestions from the NIEM community, and documenting conclusions and further actions.

Using JSON with the NIEM reference model, MPDs, and IEPs

There are three major NIEM conformance targets for which the NTAC considered support for JSON:

- **The NIEM Reference Model**. This comprises the NIEM core and domains as specified in the NIEM reference XML schema set available at niem.gov. A subset of the NIEM reference model is commonly included in an MPD.
- **MPDs, primarily IEPDs**. These are the descriptions for the content, structure, and semantics of an information exchange (or other model package such as an Enterprise Information Exchange Model [EIEM] and its Business Information Exchange Components

[BIEC]). The key normative artifacts are extension XML schemas, which include descriptions and definitions of the business content and structure specific to the scope of the MPD.
- **IEPs**. An instance of an information exchange with actual business data.

Support for JSON could be considered for any or all of these conformance targets, so using JSON does not have to be an "all or nothing" prospect.

The NIEM Reference Model and MPDs are specified using XML Schema as the primary normative artifacts. *Because JSON currently has no accepted standard for specifying allowed content and structure analogous to XML Schema, NTAC determined it would be impractical at this time to represent the NIEM Reference Model and MPDs using JSON. Therefore, our focus for 2016 is guidance for using JSON in an IEP and a specification for a NIEM-conformant JSON IEP.*

NTAC approach to using JSON with NIEM

NTAC adopted the following approach to accommodate JSON in NIEM:

1. Identify potential actions to accommodate JSON in NIEM
2. Decide which of the identified actions we think are feasible and should be pursued, and develop a roadmap
3. Explore each accepted action in detail
    a. Validate the feasibility and appropriateness
    b. Determine how to do it
    c. Write the appropriate product (guidance, specification, etc.)
    d. Consider and address tool support
4. Document a rationale for each rejected action

Steps 1 and 2 are complete, as reflected in the discussion in this document. Steps 3 and 4 are in progress.

Potential actions for using JSON with NIEM

The NTAC identified and considered the following potential actions to accommodate JSON in NIEM:

1. **Provide non-normative guidance on a way to generate a JSON instance from a NIEM-conformant XML schema and/or XML instance (based on an existing standard, specification, or practice), but the JSON instance would not be NIEM-conformant.**
2. **Provide a normative specification on a way to generate a JSON instance from a NIEM-conformant XML schema and/or XML instance, and such a JSON instance could be NIEM-conformant.**
3. **If a JSON instance could be NIEM-conformant, provide a normative specification on verifying NIEM-conformance of a JSON instance to a NIEM-conformant XML schema.**
4. Provide a normative specification for a NIEM-conformant JSON schema (based on an existing JSON Schema specification).
5. If a JSON schema could be NIEM-conformant, provide a normative specification on verifying NIEM conformance of a JSON instance to a NIEM-conformant JSON schema.

6. Provide the NIEM reference schemas as JSON schemas.
7. Develop a separate NDR for JSON (as opposed to XML).

The NTAC decided to pursue actions 1-3 for 2016. NTAC is developing a separate paper, *Potential actions for using JSON with NIEM*, with additional rationale on NTAC's pursuit of each of these potential actions.

NIEM-JSON roadmap

The NTAC developed the following roadmap to execute the accepted actions listed in the previous section:

1. **Develop and publish an executive summary and roadmap of planned NTAC actions and solutions related to JSON** (1Q2016). This serves to notify the NIEM community of our plans to support the use of JSON with NIEM and our proposed approach and timeline for doing so. The release of this paper completes the action.
2. **Develop and document a rationale for why NTAC is not pursuing actions #4 through #7** (2Q2016). This shows the NIEM community that we've fully considered the entire scope of using JSON with NIEM; it helps them understand why we've limited JSON NIEM conformance to IEPs at this time, and what events or conditions would lead us to expand the scope of JSON NIEM conformance in the future. The paper *Potential actions for using JSON with NIEM*, currently in development, will complete this action.
3. **Do action #1: Develop and publish non-normative guidance on generating a JSON instance** from an XML schema and/or XML instance (2Q2016). This allows interested organizations to begin exploring the use of JSON with NIEM in the near term, and provides an opportunity for feedback from the community based on actual experiences. This effort has started and is ongoing.
4. **Do actions #2 and #3: Develop and publish a draft normative specification (building on the initial guidance) for generating a NIEM-conformant JSON instance** from an XML schema and/or XML instance, and for verifying NIEM conformance of a JSON instance to a NIEM-conformant XML schema (4Q2016)**.** This allows organizations to implement information exchanges using JSON and assert NIEM-conformance.
5. **Make changes to the NIEM Conformance Specification to support a JSON instance conformance target** (4Q2016). This makes the NIEM Conformance Specification consistent with the normative specification for generating a NIEM-conformant JSON instance.
6. **Examine the potential for using NIEM with a code-first development approach** while still supporting NIEM goals (2Q2016). Research and evaluate options, solicit suggestions from the NIEM community, and document conclusions and further actions.
7. **Solicit feedback from the NIEM technical community as we go along.** This helps ensure that the steps we are taking benefit those who want to use NIEM and JSON together.

Summary

The NIEM Architecture Committee (NTAC) is examining the use of JSON for use with NIEM, and has developed an approach and roadmap for supporting the use of JSON with NIEM. In 2Q2016, NTAC plans to publish initial non-normative guidance for building a JSON information exchange instance

(IEP) based on a NIEM MPD. In 4Q2016, NTAC plans to develop and publish a draft specification for NIEM-conformant JSON IEPs. NTAC is also continually soliciting suggestions on using JSON with NIEM, and feedback on this paper and other NTAC papers, guidance, and specifications on using JSON with NIEM.

Comments on this executive summary and roadmap are encouraged and can be submitted on the NIEM website at https://www.niem.gov/Pages/contact.aspx.