

WS-Enumeration and OGSA: A Clash of Cultures

Mark Morgan, Global Bio Grid Group, University of Virginia

Table of Contents

Introduction.....	1
Popularity.....	1
WS-Addressing.....	2
Iteration Direction.....	2
OGSA Base Profile: The WSRF “way”.....	2
WS-Lifetime.....	3
WS-Notification.....	3
WS-BaseFaults.....	3
WS-ResourceProperties.....	3
Conclusion.....	4

Introduction

In this paper I consider WS-Enumeration [1] as an iteration service mechanism for OGSA specifications (such as ByteIO, RNS, and BES) and related OGF specifications (e.g. WS-DAI). In particular, I present reasons why WS-Enumeration is not an ideal candidate for this purpose and why an alternative specification (such as WS-Iterator) would prove superior.

This document is not meant to be an in depth comparison of WS-Enumeration and WSRF. Rather, its purpose is simply to point out elements within the WS-Enumeration specification that put it in conflict with the “OGSA-way” of the world. In depth technical comparisons of things like WS-Enumeration faults and WSRF-BaseFaults [4] are left as an excersize for the interested and motivated reader.

Popularity

The first thing to consider when looking at a potential specification for use in other specification is the popularity, ubiquity, and probable survivability of said specification. In the case of WS-Enumeration, we have a document that is still in the “submission” phase to the W3C, that has seen little action in that regard since 2006, and which seems to have been adopted by only a hand full of organizations (Microsoft and Globus stand out). Further, Microsoft itself is in the process of refining this specification. This specification, at least in the form that it currently exists in, seems to be a specification on its way out the door.

WS-Addressing

WS-Addressing [2] is the cornerstone of nearly all OGSA (and for that matter, OGF) work. If any one web service specification were to be labeled as the cornerstone or foundation upon which OGF has built its legacy, WS-Addressing would have to be that foundation. Unfortunately, due to a lengthy period of high churn in the WS-Addressing specification, a large number of versions of WS-Addressing have surfaced which, due to an unfortunate side-effect of XML namespaces, are all completely non-interoperable with one another. For better or for worse, OGSA specifications have focused on a particular instances of the WS-Addressing specification existing in the namespace <http://www.w3.org/2005/08/addressing>. Unfortunately, WS-Enumeration as it is currently defined mandates the <http://shcemas.xmlsoap.org/ws/2004/08/addressing> version. This particular problem is perhaps the biggest show-stopper leaving the would-be OGSA-WS-Enumeration implementor with the unfortunate choice of maintaining two addressing stacks, or implementing a one-off version of WS-Enumeration with an alternative addressing core.

Iteration Direction

WS-Enumeration uses a “dynamic token” mechanism for maintaining a client's position with in an iterable set. This token is considered a “black-box” of data that the client is obligated to maintain throughout the duration of a WS-Enumeration interaction. Further, each successive call on a WS-Enumeration resource results in a potentially new, or modified token that must be used in the subsequent invocation. Unfortunately, as defined, this technique permits only an iterate forward means of communication. Further, iteration must proceed sequentially without skipping, mandating that the client engage in potentially lengthy and inefficient iteration of a set of data when perhaps a more efficient binomial search might suffice. It is worth noting that WS-Enumeration does support filters which could be considered a means of “skipping” entries within an iteration context. However, this filtering mechanism presupposes that the client knows ahead of time (when the enumeration context is created) which entries to skip. One can imagine cases where the client “learns” as it traverses the enumeration about entries that can be skipped.

OGSA Base Profile: The WSRF “way”

Another bump in the road with respect to using WS-Enumeration in the OGSA stems from the existence of the OGSA Base Profiles. These profiles are documents within the OGSA which offer guidelines and rules for using existing web services technology in a consistent manner. Because different implementers and different organizations may have different “models” for how to interact with service resources, the OGSA has decided to allow multiple “renderings” of these profiles, each corresponding to a specific model of management and interaction. This is similar in concept to a language like C, C++, or Java allowing a myriad of different programming styles, thus permitting individual companies and industries to suggest/enforce specific coding styles and practices upon their developers.

At the moment, the only OGSA Base Profile in existence is the OGSA WSRF Base Profile [3] – a

profile rendering that describes interactions with grid resources using the model set forth by WSRF. Unfortunately, a number of components of WSRF are in direct opposition to WS-Enumeration.

In the following sub-sections I will indicate specifically areas of conflict between WS-Enumeration and the OGSA WSRF Base Profile.

WS-Lifetime

WS-Enumeration describes a mechanism for “releasing” WS-Enumerators which has the effect of destroying or freeing the WS-Enumeration resource. This is in direct competition with WS-Lifetime [6], a WSRF specification, mandated by the OGSA WSRF Base Profile.

WS-Notification

In the case where an enumeration is terminated unexpectedly, a WS-Enumeration endpoint is required to send a notification (when specified) to a target. Once again, this WS-Enumeration notification message competes in functional operation with an equivalent WSRF specification mandated by the OGSA WSRF Base Profile – namely WSRF-BaseNotification [5].

It is important to note at this point that the notification described above with respect to WS-Enumeration is in fact required. A cursory read of the WS-Enumeration specification might lead one to believe that this is optional given the fact that the registration of the notification endpoint is optional. However, while the client may opt to participate or not participate in the enumeration termination notification, a WS-Enumeration implementation is not free to ignore clients that do wish to receive the notification, thereby making this mechanism mandatory for WS-Enumeration implementations. This particular pattern of optional client behavior but mandatory service implementation behavior is common in WS-Enumeration and will pop up from time to time in this paper.

WS-BaseFaults

The WS-Enumeration document describes in detail the mechanisms in place for transmitting fault or error information back to clients. This faulting mechanism is actually described in two different formats; one for SOAP 1.1 implementations, and one for SOAP 1.2 implementations thus granting WS-Enumeration some amount of duality between different implementation bases and client bases. Unfortunately, both of these faulting mechanisms conflict with the WSRF-BaseFaults [4] faulting format, mandated by the OGSA WSRF Base Profile.

WS-ResourceProperties

WS-Enumeration describes a mechanism for clients to obtain status information, or metadata about target enumerations. This metadata essentially amounts to what WSRF would consider to be resource

properties of that enumeration. Unfortunately, once again, the differing syntax and interaction mechanisms described by WS-Enumeration as compared to WSRF-ResourceProperties [7] make WS-Enumeration unsuitable for OGSA WSRF Base Profile implementations.

Conclusion

Programming model and paradigm is an often undervalued component of software design. Often, which model an application is built using is not nearly as important as a conscious, “good intentions” adherence by programmers and software architects to that model. Consistent design within the overall model of a programming system leads to more readable, more reusable, and more cohesive system design. In the absence of this strict adherence to programming model, systems grow cancerously, often forcing developers to hammer “square” software components into “round” holes in software systems.

OGSA's adoption of a *** Base Profile** mechanism for OGSA specification design is equivalent in design intent to a set of coding standards that a software development company might adopt. Due to the OGSA Base Profile “rendering” options, it may be the case that in the future an OGSA Base Profile will exist for which WS-Enumeration is an ideal candidate when a specification mandates the use of iterable content. However, in the case of the OGSA WSRF Base Profile, it seems clear that WS-Enumeration is simply too foreign to fit cleanly with other OGSA specifications.

Instead, what one would like for an OGSA WSRF Base Profile referent iterator is a simple iteration port type that allowed for both forward and backward traversals and which rendered useful, core resource operations such as destruction, notification, faulting, and metadata management using available WSRF mechanisms like WS-Lifetime, WS-BaseNotification, WS-BaseFaults, and WS-ResourceProperties.

References

- [1] Alexander, J., Box, D., Cabrera, L.F., et. al., *Web Services Enumeration (WS-Enumeration)*, <http://www.w3.org/Submission/2006/SUBM-WS-Enumeration-20060315>, 15 March 2006.
- [2] Gudgin, M., Hadley, M., Rogers, T. (ed.), *Web Services Addressing 1.0 – Core (WS-Addressing)*, 9 May 2006, <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509>
- [3] Foster, I., Maguire, T., Snelling, D., *OGSA WSRF Basic Profile 1.0*, <http://www.ggf.org/documents/GFD.72.pdf>, GFD.72, 1 May 2006.
- [4] Tuecke, S., Liu, L., Meder, S. (ed.), *Web Services Base Faults 1.2 (WS-BaseFaults)*, OASIS Standard, 1 April 2006, http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf.
- [5] Graham, S., Hull, D., Murray, B. (ed.), *Web Services Base Notification 1.3 (WS-BaseNotification)*, OASIS Public Review Draft 02, 28 November 2005, http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf.
- [6] Srinivasan, L., Banks, T. (ed.), *Web Services Resource Lifetime 1.2 (WS-ResourceLifetime)*, OASIS Standard, 1 April 2006, http://docs.oasis-open.org/wsrf/wsrf-ws_resource_lifetime-1.2-spec-os.pdf.
- [7] Graham, S., Treadwell, J. (ed.), *Web Services Resource Properties 1.2 (WS-*

ResourceProperties), OASIS Standard, 1 April 2006, http://docs.oasis-open.org/wsrp/wsrp-ws_resource_properties-1.2-spec-os.pdf.