Interoperability Systems

Todd Malone
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
malon153@morris.umn.edu

Keywords

ACM proceedings, LaTeX, text tagging

1. ON INTEROP: FOCUS AND GOALS

This paper will discuss interoperability of programming languages, why it may be important or required, and the difficulty involved[11, 4]

I'll look at some general ideas of what it means to be interoperable and how to accomplish that.[6, 10]

I will also look at two particular methods for achieving interoperability: Virtual machines and markup languages. Each of these will have a particular example associated with it.

- Virtual machines will have .NET or the JVM [8, 5]
- Markup languages will have the the Starlink system [2] and potentially FML [1]

I have a few papers with uncertain utility[7, 9, 3], but they may still be useful. Apart from these, I will likely be adding some Wikipedia sources for certain background knowledge (in the areas of virtual machines and markup languages).

2. REFERENCES

[1] G. Acampora. Fuzzy markup language: A xml based language for enabling full interoperability in fuzzy systems design. In G. Acampora, V. Loia, C.-S. Lee, and M.-H. Wang, editors, *On the Power of Fuzzy Markup Language*, volume 296 of *Studies in Fuzziness and Soft Computing*, pages 17–31. Springer Berlin Heidelberg, 2013. ***ML paper. This is much more concerned with fuzzy systems and hardware interop than it is with cross-language interop. However, it does talk of using an ML to divorce fuzzy controllers from both hardware and software, and a translator to convert it into the correct language. This could be useful.*.**

[2] Y. Bromberg, P. Grace, and L. ReÌĄveilleÌĂre. Starlink: Runtime interoperability between heterogeneous middleware protocols. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 446–455, 2011. ***Markup Language paper. This details a system built to achieve interop between any two systems, with MLs as the translational language. This will be my major example when talking about markup languages*.**

[3] C. Chen, D. Brown, C. Sconyers, G. Vachtsevanos, B. Zhang, and M. Orchard. A .net framework for an integrated fault diagnosis and failure prognosis architecture. In *AUTOTESTCON, 2010 IEEE*, pages 1–6, 2010. ***Unused VM paper. Most of this isn't useful, but has a description of the .NET framework. Might be more useful to just look up the wikipedia*.**

[4] D. Chisnall. The challenge of cross-language interoperability. *Queue*, 11(10):20:20–20:28, Oct. 2013. ***Contains a short section on why VMs are not a cure-all interop method. This might actually be useful as a way of transitioning between VMs and MLs, since MLs cover pretty much the exact area VMs don't*.**

[5] J. Hamilton. Language integration in the common language runtime. *SIGPLAN Not.*, 38(2):19–28, Feb. 2003. ***Virtual machine paper. Discusses the methods the .NET framework uses to promote interoperability. Specificlaly metadata tracking, common type specifications, and other common requirements. Probably the most useful thing I've come across regarding VMs, so this will likely be my primary source for that section*.**

[6] N. Ide and J. Pustejovsky. What does interoperability mean, anyway? toward an operational definition of interoperability for language technology. In *Proc. 2nd Int. Conf. Global Interoperability Lang. Res*, 2010. ***Background. A very general exploration of the definitions and requirements of interoperability. Primarially Talks about metadata as the interop method*.**

[7] L. Kats and E. Visser. Encapsulating software platform logic by aspect-oriented programming: A case study in using aspects for language portability. In *Source Code Analysis and Manipulation (SCAM), 2010 10th IEEE Working Conference on*, pages 147–156, 2010. ***Unused Background paper. Mainly***

UMM CSci Senior Seminar Conference, December 2013 Morris, MN.

involves seperating languages from platform dependency. While related, I don't think this will be an area that I focus on.

[8] W. H. Li, D. R. White, and J. Singer. Jvm-hosted languages: they talk the talk, but do they walk the walk? In *Proceedings of the 2013 International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools*, pages 101–112. ACM, 2013. ***A study quantifing the differences languages run on the JVM from Java itself. Shows that JVM languages, even languages not built specifically for it, really do behave differently than Java (and each other) at runtime.***

[9] J. Matthews and R. B. Findler. Operational semantics for multi-language programs. *ACM Trans. Program. Lang. Syst.*, 31(3):12:1–12:44, Apr. 2009. ***Long, but talks about high-level (more abstract?) considerations in language interop. May be useful, if I have time to read it.***

[10] P.-M. Osera, V. Sjöberg, and S. Zdancewic. Dependent interoperability. In *Proceedings of the Sixth Workshop on Programming Languages Meets Program Verification*, PLPV '12, pages 3–14, New York, NY, USA, 2012. ACM. .

[11] D. S. V. Sujala D Shetty. Interoperability issues seen in web services. *IJCSNS International Journal of Computer Science and Network Security*, 9:160–169, August 2009. ***Background paper. Talks about output mismatch between Java and .NET web clients. This includes things like NULL, primitives, and precision, and a little involving objects.***