Douglas C. Schmidt
Phone: (314) 935-4215

August 22, 1999

Edward Lasowski
Dept. of Comp. Sci. and Eng.
University of Washington
Box 352350
Seattle, WA 98185-2350

Dear Dr. Lasowski,

I am writing this letter to support Dr. Bjarne Stroustrup's nomination for a Turing Award. I have used C++ actively for over a decade to support my research on high-performance computer networking protocols and real-time middleware. I have interacted with Bjarne many times during this period. Therefore, I am quite familiar with his outstanding research talents and accomplishments and his impeccable personal integrity. During the many years that I've known him, Bjarne has consistently impressed me as being a highly perceptive and dedicated researcher whose research has had substantial impact on R&D in the computer software field. His work attacks hard, real world problems, and yields substantive results that neither oversimplify nor promote superficial solutions. This is an all too rare combination in the research community.

Over the 12 years I've worked with C++, it has evolved from a research project at AT&T Bell Labs to become one of the most widely used object-oriented (OO) programming languages for software research and development. During this period, many other programming languages, such as Ada, Eiffel, Java, Modula 3, Oberon, and Objective-C, also made their debut. Some of these languages gained strong followings in certain domains, such as military applications and user interfaces. However, I believe that C++, more than any other programming language, has helped to improve software development practices in my research area during the past decade.

My research focuses on the patterns, implementation, and experimental analysis of techniques that facilitate the development of high-performance and real-time distributed object computing systems running over high-speed networks and embedded backplanes.[1] I have over a decade of experience conducting OO communication systems research using C++. I am the chief architect and developer of ACE and TAO, which are widely used, freely-available C++ middleware frameworks that contain a rich set of components that implement design patterns for high-performance and real-time communication systems. In conjunction with my academic research, I have successfully used ACE, TAO, and C++ on dozens of large-scale commercial projects at many companies, including Ericsson, Siemens, Motorola, Lucent, Lockheed Martin, Boeing, and SAIC. These projects involve telecommunications systems, medical imaging systems, real-time avionic systems, and distributed interactive simulation

[1] More details on my background and experience are available at www.cs.wustl.edu/∼schmidt/research.html.

systems.

Based on my extensive experience with both research and development, I think it's fair to say that without C++ and its particular paradigm of *efficient OO programming*, it would not have been possible to inculcate advanced software principles, patterns, and processes into demanding high-performance and real-time systems. Most importantly and uniquely, C++ combines both abstraction *and* performance. In particular, while C++ retains C's emphasis on efficient run-time execution, its use of data abstraction, inheritance, dynamic binding, and parameterized types substantially alleviates much of the tedious and error-prone aspects of lower-level programming for real-time and high-performance systems.

In retrospect, it is clear that a key reason for C++'s success as a language, and more generally, the success of OO programming in research and commercial projects, stems from the substantial contribution of Bjarne Stroustrup. Bjarne has spent countless hours helping others succeed with C++ and OO programming through postings on USENET, tutorials at conferences, and insightful articles for journals and magazines. He's also written many of the classic books on C++ and OO programming. His books go far beyond the conventional coverage of tactical programming language features and explore the key strategies and patterns necessary to develop complex software systems effectively.

What is particularly impressive to me is Bjarne's level of integrity and technical leadership within the OO programming community. Too often, advanced programming skills are devalued by academics and researchers who look towards Process and Methodology as "silver bullets" to address complex software development challenges. Unfortunately, researchers and developers who actually write complex software systems understand all too well that skilled programmers are essential to success. To his credit, Bjarne has long emphasized the need to recognize and reward programming skills. In all of the many keynote address or invited talk I've seen Bjarne give over the years, he has cautioned against seeking magic elixirs and panaceas at the expense of pragmatic solutions and technical substance.

Bjarne's emphasis on pragmatism in C++ is evident in language features like inline functions and parameterized types. These features make it possible to write highly flexible OO programs whose run-time efficiency, scalability, and predictability, is equivalent to those coded manually using lower-level languages, such as C or assembly language. These features and design choices made it possible for the R&D projects I've been involved with to use C++ for such a wide spectrum of applications, ranging from resource-constrained embedded control systems to large-scale telecommunication software systems.

Bjarne's emphasis on technical substance also has helped shape many key C++ features and design decisions. For instance, one of the most important decisions he was involved with was shepherding the Standard Template Library (STL) into the ISO/ANSI C++ standard. STL provides C++ programmers with powerful collections and generic algorithms for creating and reusing highly efficient "pluggable" components. This power does not come for free, however, since STL requires developers to invest significant time to master the generic programming paradigm. Bjarne's willingness to personally help C++ developers internalize the appropriate programming patterns and idioms needed to leverage the full power of C++ made the addition of STL to the C++ well worth the risk.

In today's software environment, mastery of C++ and OO programming are just two of many skills that researchers and developers must have to succeed. In addition, they must be highly proficient with cross-platform development environments, multi-threading, object-oriented databases, GUI frameworks, distributed object computing, and the World Wide Web. I'm optimistic that Bjarne's research at AT&T Labs will continue to provide tomorrow's programmers with the technical leadership, strategies, tactics, and tools they need to succeed.

In summary, Bjarne possesses an impressive array of capabilities that make him particularly worthy of receiving a Turing Award. He is a highly imaginative and dedicated researcher who for over twenty years has scoped out and successfully conducted R&D efforts that have had a substantial technical impact on the computer software field. From our close interaction together the past decade, Bjarne has amply demonstrated his research capabilities, technical leadership, and personal integrity. Therefore, I enthusiastically recommend Bjarne for a Turnig Award.

If I can provide any additional information, please feel free to contact me.

Sincerely,

Douglas C. Schmidt
Associate Professor
http://www.cs.wustl.edu/~schmidt/
Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis, Missouri 63130-4899
(TEL): (314) 935-7538
(FAX): (314) 935-7302