Issues in the Certification of Reusable Parts

John C. Knight
Department of Computer Science
University of Virginia
Thornton Hall
Charlottesville, VA 22903

knight@virginia.edu

Abstract

A substantial difficulties that is limiting reuse is a lack of perceived quality in the artifacts being reused. Availability of a part with suitable functionality is not sufficient if the prospective user does not trust the part and refuses to use it as a result. We present a definition of certification to deal with this situation and propose to exploit this definition to permit savings during testing and maintenance.

Keywords: software reuse, part certification, software reliability

1 Issues in the Certification of Reusable Parts

We claim that one of the substantial difficulties that is limiting reuse is a lack of perceived quality in the artifacts being reused. Although a software engineer may have a reuse library available, there is frequently a reluctance to use it because of concerns about part quality. Essentially, the engineer feels that without a lot of knowledge of a part, he or she would be better off rebuilding it. That lack of perceived quality is a detractor from reuse is an observation based only on anecdotal evidence but appears to be the software-reuse manifestation of the "not-invented-here" syndrome.

The adjective *certified* is sometimes used to describe parts that have been tested in some way prior to entry into a library (e.g., [Lenz 87]. Testing parts prior to their insertion into a reuse library is often claimed to be a productivity advantage. There is the vague expectation that building software from tested parts will somehow make testing simpler or less resource intensive, and that products will be of higher quality [Bass 87, Knig 92, Lenz 87]. Despite the various discussions of testing and reuse, the term certified is not formally defined in the reuse literature.

We are engaged in a research program that is addressing the issue of certifying reusable parts. We advocate the development of software by reuse with the specific intent of establishing as many

of the required properties in the final product as possible by depending upon properties present in the reusable parts. For this goal to succeed, a precise definition of certification of reusable parts is required. Given the informal notions of certification that have appeared, it is tempting to think that a definition of certification should be in terms of some test metric or similar. The major difficulty with this approach, no matter how carefully applied, is that any single definition that is offered cannot possibly meet the needs of all interested parties. A second difficulty is that by focusing on a testing-based definition, other important aspects of quality, such as efficient execution performance or ease of maintenance, are omitted from consideration. These difficulties have lead us to establish the following definitions:

- **Definition: Certification Instance**: A certification instance is a set of properties that can be possessed by the type of part that will be certified according to that instance.
- **Definition: Certified Part**: A part is certified according to a given certification instance if it possess the set of properties prescribed by that instance.
- **Definition: Certification**: Certification is the process by which it is established that a part is certified.

An important byproduct of this precise definition of certification is that it provides a mechanism for *communication* about part quality between the developer of a part and users of the part. Users no longer have to question the quality of parts - certification describes for the prospective user exactly what can be expected of a part. When developing a part for placement in the library, it is the developer's responsibility to show that the part has the properties required for that library. When using a part, it is the user's responsibility to enquire about the precise set of properties that the part has and ensure that they meet his or her needs.

These definitions appear to be of only marginal value because the prescribed properties are not included. However, it is precisely this aspect that makes the definitions useful. The definitions have three very valuable characteristics:

1. **Flexibility:** As many different certification instances can be defined as are required.
2. **Generality:** Nothing is assumed about the *type* of part to which the definitions apply.
3. **Precision:** Once the prescribed property list in the certification instance is established, there is no doubt about the meaning of certification.

The properties included in a specific instance of certification can be anything relevant to the organization expecting to use the certified parts. However, since preparation of reusable parts is a major capital undertaking, it is inappropriate to include properties that are not essential. The opposite circumstance is also a factor. If establishing a necessary system characteristic is facilitated by the parts in the reuse library having a certain property, then that property had better be included in the certification instance.

Thus the key to the definition of any specific certification instance is the use to be made of the properties in the definition. The *only* justification for the inclusion of a particular property in a

specific certification instance is that possession of that property by parts in a library contributes to the establishment of useful characteristics in systems built from those parts.

We now have a general definition of certification for reusable parts and a conceptual approach to developing specific definitions as required. The key issues to be addressed in the area of certification are:

1. What *system* properties are common and of sufficiently high value that supporting them in a reuse development environment is cost effective?

2. What techniques are required to permit the maximum exploitation of the properties of parts in the establishment of the properties of a system?

3. What library structures are required to store the relatively complex entity formed from a part and an associated set of properties?

4. Is there value in permitting library users to search based on both part semantics and required certification properties?

5. How does certification according to the approach outlined here affect the economic models of reuse?

6. Since it is clear that properties of parts based on testing will be included in certification definitions, what issues are raised in the area of testing parts? The difficulty of testing artifacts such as Ada [Ada 83] generic units immediately comes to mind.

7. What effect does the adaptation of reusable parts have on the definition of certification and its exploitation?

8. Can the notion of certification summarized above be applied successfully to parts other than source code and it is similarly advantageous?

9. What would the instantiation of certification look like for requirements or test-plan parts?

For software reuse to succeed in delivering a substantial improvement in programmer productivity requires progress in a number of areas. Part certification is an important one.

References

[Bass 87] Bassett, P.G., "Frame-Based Software Engineering", *IEEE Software*, July, 1987.

[Knig 92] Knight, J.C., "Certification of Reusable Parts" , submitted to *ISQE 92*, March, 1992.

[Lenz 87] Lenz, M., H.A. Schmid, and P.F. Wolf, "Software Reuse Through Building Blocks" , *IEEE Software*, July, 1987.

[Trac 87] Tracz, W., "Software Reuse: Motivators and Inhibitors", *Proceedings of COMPCON S'87*, 1987.

[Ada 83] U.S. Department of Defense, Ada Joint Program Office, *Reference Manual For The Ada Programming Language*, ANSI/MIL-STD-1815A, January, 1983.

2 About the Author

John C. Knight is an associate professor of computer science at the University of Virginia. Before joining the University of Virginia in 1981, he was with NASA's Langley Research Center. His research interests center on developing techniques for building software for safety-critical systems. To that end, he is undertaking a research project in certification of reusable parts with a view to exploiting reuse to support software reliability.