

Gearbox: Truly reusable robot software

Geoffrey Biggs
Intelligent Systems Research Institute
National Institute of Advanced Industrial
Science and Technology (AIST), Japan
geoffrey.biggs@aist.go.jp

Alexei Makarenko, Alex Brooks, Tobias Kaupp and
Michael Moser
ARC Centre of Excellence for Autonomous Systems (CAS)
The University of Sydney, Australia
{a.makarenko,a.brooks,t.kaupp,m.moser}@cas.edu.au

Despite many years of effort and development on the part of a large number of researchers, true software reuse has yet to be realised in robotics software. Several approaches have been tried, including developing an entire software framework from scratch (including the communications system) [1], using off-the-shelf software to provide the middleware [2], creating bridges between frameworks [3], and defining standards to allow different robot software frameworks to interact [4]. However, in every Robotics Software System (RSS) project, there is inevitably a large quantity of underlying functionality that is common to all RSS projects but rewritten from scratch or copied from other projects and then maintained separately. This monolithic approach has led to a situation where reusability is unnecessarily restricted.

The Gearbox project (<http://gearbox.sourceforge.net>) is an attempt at solving the reusability problem by stepping away from robot software frameworks. The guiding principle of Gearbox is that the solution to truly reusable robot software is to separate *functional* software from *integration* software. Gearbox focuses on the former by providing a set of individual software implementations that provide the functional software for robotics, such as hardware drivers and algorithm implementations. These are realised as libraries that can be freely used in any way a developer sees fit, including creating components based on them for use in frameworks. The hope is that by not limiting each library to a specific framework, they will be usable in all frameworks. It is important to emphasise that Gearbox does not consider frameworks to be inherently bad, but rather that software reusability needs to occur on more than one level. Gearbox was inspired by the concept of thin frameworks proposed in [5] and the UNIX model of interacting libraries and programs. The primary goal is the provision of a collection of usable shared libraries of robot functionality that can be easily installed on a system and used in whichever way the developer chooses.

The important principles of Gearbox can be summarised as:

- *Gearbox is not a framework.* Gearbox does not contain any integration code (such as middleware), and neither competes with any existing or future frameworks nor removes the need for them.
- *Freedom of implementation* is particularly important. Unnecessary restrictions upon developers (such as specifying that a certain API or programming language must be used) would discourage contributions. However, a common build system is imposed across the entire collection

to ensure ease and consistency of installation.

- *Be applicable to robotics.* The scope of Gearbox must be limited to avoid creating a bloated collection.
- *Libraries must be usable.* Libraries should have a consistent, well-documented API.
- *Alternative implementations are acceptable.* Having two libraries that implement the same functionality is a benefit if they each have advantages over the other.
- *Open source.* While each developer of a library is free to choose the license, it must be open-source.
- *Peer review.* To ensure a reasonable standard, contributed libraries are reviewed and discussed on a public mailing list prior to acceptance into Gearbox.

Gearbox has advantages for researchers, framework users, functional library developers and framework developers. It will create wider reuse of functional robot software, ease maintenance of this software, ease the development of new RSS projects, and improve the comparison of RSS projects (when the underlying implementation is identical, the differences due to the framework itself become more obvious). These benefits outweigh the extra time and effort required to develop a suitable library for Gearbox over that of developing for a single RSS project.

Gearbox is still in the early stages of growth, currently containing eight libraries, including laser scanner and GPS drivers. It is used by three frameworks: Orca, Player and OpenRTM-aist. As it gains more libraries and its available functionality grows, the advantages to robot developers of all kinds will grow significantly. Gearbox now needs contributions by other developers of robot software in order to grow.

REFERENCES

- [1] R. T. Vaughan, B. P. Gerkey, and A. Howard, "On device abstractions for portable, reusable robot code," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, vol. 3, 2003, pp. 2421–2427.
- [2] A. Brooks, T. Kaupp, A. Makarenko, S. Williams, and A. Oreback, "Orca: a component model and repository," in *Principles and Practice of Software Development in Robotics*, D. Brugali, Ed. Springer, 2007.
- [3] C. Cote, D. Letourneau, F. Michaud, J.-M. Valin, Y. Brosseau, C. Raievsky, M. Lemay, and V. Tran, "Code reusability tools for programming mobile robots," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, vol. 2, 2004, pp. 1820–1825.
- [4] (2006) The OMG Standard Fall 2006 Issue. [Online]. Available: http://www.omg.org/news/newsletters/OMG-Standard-Fall_2006.pdf
- [5] A. Makarenko, A. Brooks, and T. Kaupp, "On the benefits of making robot software frameworks thin," in *Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2007.