

Can open source ever make an impact in robotics?

Herman Bruyninckx

—
Department of Mechanical Engineering
K.U.Leuven, Belgium
—

ICRA 2009, Kobe, Japan. 17 May, 2009

Overview

- ▶ Context
- ▶ Current status
- ▶ What's missing?
- ▶ Risks
- ▶ How to proceed?
- ▶ Conclusions

Context

- ▶ My personal and professional ICT is 99% FOSS since 1994.
(FOSS = *Free and Open Source Software*)
- ▶ I have consulted for dozens of companies, governments, schools. . . about FOSS.
- ▶ I started www.orocos.org in 2000, with industrial funding.
- ▶ I'm actively promoting FOSS in (academic & industrial) robotics.
- ▶ I believe in a FOSS [infrastructure](#) with commercial [applications](#) and [services](#) on top.

Current status in robotics FOSS

- ▶ **Lots** of FOSS projects & enthusiasm.
- ▶ **Lots** of money: funding agencies love FOSS.
- ▶ **Reasonable** amount of robotics functionality available...
...but important ones still (mostly) missing:
sensor based motion control, interpreters for
commercial robot programming languages
(KRL, RAPID,...)
- ▶ Needs in robotics are **higher** than in most other
domains: robotics is all about **integrating**
several domains in one single **system**!

Current status in general FOSS

- ▶ **Successful** FOSS projects **invariably** have (large) support from **(large) companies**: Linux, Qt, Eclipse, Firefox, Apache, MySQL, ...
(**No** robotics company is significantly funding any robotics FOSS...)
- ▶ ...and are built on top of **“open” standards** (UNIX/POSIX, HTML, Java, SQL,...)
(**No** real robotics SW standards exist...)
- ▶ **Vertically integrated** fully FOSS software stacks exist: LAMP (Linux/Apache/MySQL/PHP), Plone, ...

What's missing in robotics? —Interoperability—

- ▶ No two FOSS projects in robotics work **seamlessly** together (yet).
- ▶ This is a problem of the whole **robotics domain**, not specifically of FOSS!
- ▶ No **standards** exist at the fundamental lower levels: position and motion representation; motion control; 3D world representation; . . .

What's missing in robotics?

—Consolidation—

- ▶ Too many reinvented wheels:
 - ▶ robotics “middleware” projects.
(Why not use mature, commercially backed projects from telecom, e.g. OpenSAF?)
 - ▶ 3D visualisation.
(Why not all work with Blender?)
- ▶ Let's finally agree on the **semantics** of mature concepts (cf *standards* mentioned above).
⇒ **ontology** of robotics badly needed!

What's missing in robotics?

—Toolchain—

- ▶ How to support the integration of software from different sources?
- ▶ Other domains invest in “Model-Driven Engineering”:
 - ▶ Automotive: [AUTOSAR](#).
 - ▶ Control: Simulink, Scicos,...
 - ▶ Computer Science: UML,...
 - ▶ Embedded systems: formal verification,...

Common infrastructure: www.eclipse.org!

What's missing in robotics?

—Education—

PhD students/R&D engineers don't know:

- ▶ the real differences between:
 - ▶ **Classes** (Object-oriented)
 - ▶ **Components** (Component-based)
 - ▶ **Services** (Service-Oriented Architecture)
- ▶ how to design **loosely coupled** SW.
- ▶ how to design SW that is **reusable** and **benchmarkable**.
- ▶ how **to compose** a **system** out of lots of heterogeneous components.
- ▶ how to deal with **realtime** requirements.

Risks of failure for FOSS projects

- ▶ 99% of FOSS developers in robotics are **not** working towards solving the *missing* items.
- ▶ no feeling for **real industrial needs**.
- ▶ **versioning hell**. (Too difficult to integrate versions of different, independently evolving SW projects.)
- ▶ **monolithic** SW stacks.
- ▶ **de facto** proprietary “standards”: The MathWorks, National Instruments, Microsoft.
- ▶ **too big egos** of FOSS developers. . .

How to proceed?

—Most important next steps—

- ▶ **Ontology!**
- ▶ Hook into FOSS **infrastructure** projects with large **critical mass**: **Blender**, **Eclipse** and **OSGi**.
- ▶ Bridge the **East-West gap**: OpenRTM (Japan) and OPROS (Korea) are not known/accessible by Westerners.
- ▶ **Intensive, constructive criticism** for the enormous efforts by **Willow Garage**.
- ▶ Yearly “FOSS in robotics” **conference**.

Conclusions

- ▶ **Extrapolation** from the past predicts a very unsuccessful future for FOSS in robotics. . .
- ▶ Larger **industry-involvement** is needed, but should come with **no-strings-attached!**
- ▶ Major **non-functional** step to be realized: **interoperability!!!**
- ▶ Major **technical** step to be realized: to educate our community about **loosely coupled designs** and **component-based development**.
- ▶ Major **strategic** step to be realized: to integrate with **Eclipse** and **Blender!**