Modeling Software Systems in Experimental Robotics for Improved Reproducibility

A Case Study with the iCub Humanoid Robot
Research on autonomous robots achieved considerable progress over the recent years.

Reported results are typically validated through experimental evaluation or demonstrated live at robotics competitions. [ DARPA Challenge, RoboCup, …]

Amongst other reasons, this is due to: „off-the-shelf“ robots [ iCub, NAO, … ] and vivid open source communities providing reusable building blocks, e.g. ROS, Orocos, OPRoS, Yarp, yarp-wholebodyinterface, GURLS, … ;)

Publicly available data sets are used to improve benchmarking procedures, i.e., Rawseeds Project. [1]
» Numerous studies have been performed over the past 15 years, but one of the hallmarks of science has yet to be achieved: results at present are hardly ever reproducible by other research groups. « [2]
Methodological issues that prevent reproducibility of robotic system experiments. [3]

This includes the frequently neglected impact on experiments caused by the relationship between individual components and the whole system — in component-based systems for instance.

Question: how do publications need to be written and published in order to improve reproducibility?
Systems in Experimental Robotics | **Aspect Overview**

![Diagram showing the relationship between technical aspects, engineers, interaction, system, software, data, configuration, execution protocol, documentation, deployment, evaluation, and method documentation.](image-url)
Information retrieval and aggregation: artifacts are often distributed over different locations, thus already the discovery, identification and aggregation of all required artifacts is difficult.

Semantic relationships: which specific versions (master/v. 133.7) of software components were in use for a particular study?

Software deployment: Most current systems are realized using a component-based architecture. They do not necessarily use of the same build infrastructure CMake, Catkin, maven, etc., binary deployment mechanism and execution environment.

Experiment testing, execution and evaluation: Advanced robotics experiments require significant efforts spent on system development, integration testing, execution, evaluation and preservation of results. This is particular costly if many of these tasks are carried out manually. Crucial run time parameters and component configurations are often omitted or not documented properly.
To tackle these issues we introduce an approach for reproducible robotics experimentation based on an integrated software toolchain for system developers and experiment designers.
System Model
Systems in Experimental Robotics | Technical Realization Overview

Recipes: Java, C/C++, Python, Experiments

Web-based catalog (Systems and related artifacts)
Systems in Experimental Robotics | Technical Realization Recipes

```
{
    "name": "yarp",
    "templates": [ "cor-lab", "cmake-opp" ],
    "catalogue.component.title": "Yet Another Robot Platform (YARP)",
    "catalogue.component.nid": "1274",
    "variables": {
        "description": "YARP is plumbing for robot software. It is a set of libraries, protocols, and tools to keep modules and devices cleanly decoupled.,"
        "keywords": [ "middleware", "integration", "robotics", "opp" ],
        "repository": "https://github.com/robotology/yarp.git",
        "branches": [ "master", "v2.3.63" ],
        "git.shallow?": false,
        "extra-provides": [ "cmake", "yarp", "2.3.63" ],
        "cmake.options": [ "CMAKE_BUILD_TYPE=Release",
                            "CREATE_SHELL=ON",
                            "CREATE_LIB мат=ON",
                            "CREATE_OPTIONAL_CARRIERS=ON",
                            "CREATE_SHARED_LIBRARY=ON",
                            "ENABLE_yarpcon_proto=carrier=ON",
                            "@{next-value}[]" ]
    }
}
```
Systems in Experimental Robotics | **Technical Realization Recipe (Experiment)**
Systems in Experimental Robotics | **Technical Realization Experiment Protocol**
ity. However, the sponsorship of an automated build infrastructure and tools to automatically create build jobs (cf. ROS-bloom and Section 2.2) reduces the amount of expert knowledge and is thus also considered beneficial. In contrast to our approach, ROS and iCub distributions can be installed via source builds (not recommended as stated in the ROS wiki) but also via binary distributions that simplify and speed up installation time. On the other hand, binary packages often raise typical issues such as requiring root permissions for installation, the install prefix is fixed and creating binary packages for diverse operating systems and flavors is a huge effort. With respect to build systems both ecosystems are based on CMake, which facilitates cross-platform compatibility, but also, in contrast to CITK, restricts the number of integrable third-party build tools. This is especially crucial because robotic systems/experiments often incorporate artifacts from more than one ecosystem. Finally, experiment specification, orchestration, automated execution and evaluation is not supported by either ROS or the iCub infrastructure.

In order to verify our results, please visit: iCub Ball Tracking-nightly

5 Conclusion

We introduced an approach for reproducible robotics experimentation based on an integrated software toolchain for system developers and experimenters. It combines state-of-the-art technologies into a consistent process that facilitates the reproduction of robotic systems and experiments. We briefly outlined the replication process for a simulation experiment and discussed the benefits of the approach in comparison to well-known robotics ecosystems and their support for reproducible experimentation. Future work will focus on providing the complete toolchain as open source to the community, extending the build generation with
Systems in Experimental Robotics | Step by Step

Visit system version on catalog website
BROWSE LINKED ARTIFACTS

Required Component Versions:
- Finite State Machine Based Testing (FSMT)-0.16
- Data set iCub Ball Tracking-095402
- iCub Main-v1.1.15
- Yet Another Robot Platform (YARP)-v2.3.63
- iCub Contrib-master
- Yet Another Robot Platform (YARP) Python Bindings-v2.3.63
- FSMT Experiment iCub Ball Tracking-master
- PySOXML-v0.8.3

Linked Distribution Recipe:
- icub-nightly

Linked Experiments:
- iCub Ball Tracking-nightly

Browse linked artifacts: Publications, Component Versions, Data Sets, etc...
Local & Global CI

Systems in Experimental Robotics | **Step by Step**

```
cd $prefix
wget --no-check-certificate https://toolkit.cit-ec.uni-bielefeld.de/dist/jenkins.tar.gz
```

Download pre-packaged CI Server
Extract & simply start it

cd $prefix/jenkins
./start_jenkins
Find ready to use local CI Server
cd $prefix
mkdir dist
cd dist
git clone https://opensource.cit-ec.de/git/citk .
Invoke job-configurator with desired distribution

```
```
Start build-flow job, the rest is orchestrated automatically, system is fully built (repeatable via single click)
Visit linked experiment in the web catalog and review reference data
Invoke specified command (listed in catalog) and the experiment is executed locally.
Demo
Systems in Experimental Robotics | Static Demo

https://vimeo.com/112005754
Modeling of artifacts required for system replication and experiment execution

‘Easy to use’ (yet to be proven) system deployment strategy for local and ‘global’ use case

Inherent CI paradigm for software and experiment provenance

Targets software developers, experiment designers and interested researchers / reviewers

Enables early integration of experiment designers — R, Matlab scripts in the loop

Ideal for early testing with Simulation environments (as shown in the demo)

Browsable web catalog of semantically linked research artifacts
We modeled **HARDWARE** but we still need to provide a working use case

Faster ‘shipping’ methods, i.e., for reviewers only, using VM images or Linux containers like Docker

Automated data annotation, i.e., post experiment data processing like annotation of video material not supported

Build/Exceution in the cloud (sorry) currently not investigated

Support for more than 2 Linux flavors, Mac OSX, probably Windows too

... and probably much more
Credits

Thanks Lorenzo for the invitation and Vadim for the technical support & the iCub friends for the _ready to use_ software stack!
Thank you!

Links

Web Catalog: https://toolkit.cit-ec.de
Distribution Project: https://opensource.cit-ec.de/citk
Video Material: http://vimeo.com/groups/citk

Affiliation

Florian Lier, Sven Wachsmuth and Sebastian Wrede are with Cognitive Interaction Technology — Center of Excellence, Bielefeld University (www.cit-ec.de)
Sebastian Wrede is also with The Research Institute for Cognition and Robotics — CoR-Lab (www.cor-lab.de)
References

HANDS ON SESSION — Let’s integrate?