CITEC

Cognitive Interaction Technology

Exzellenzcluster

Universität Bielefeld

CoR-Lab

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]

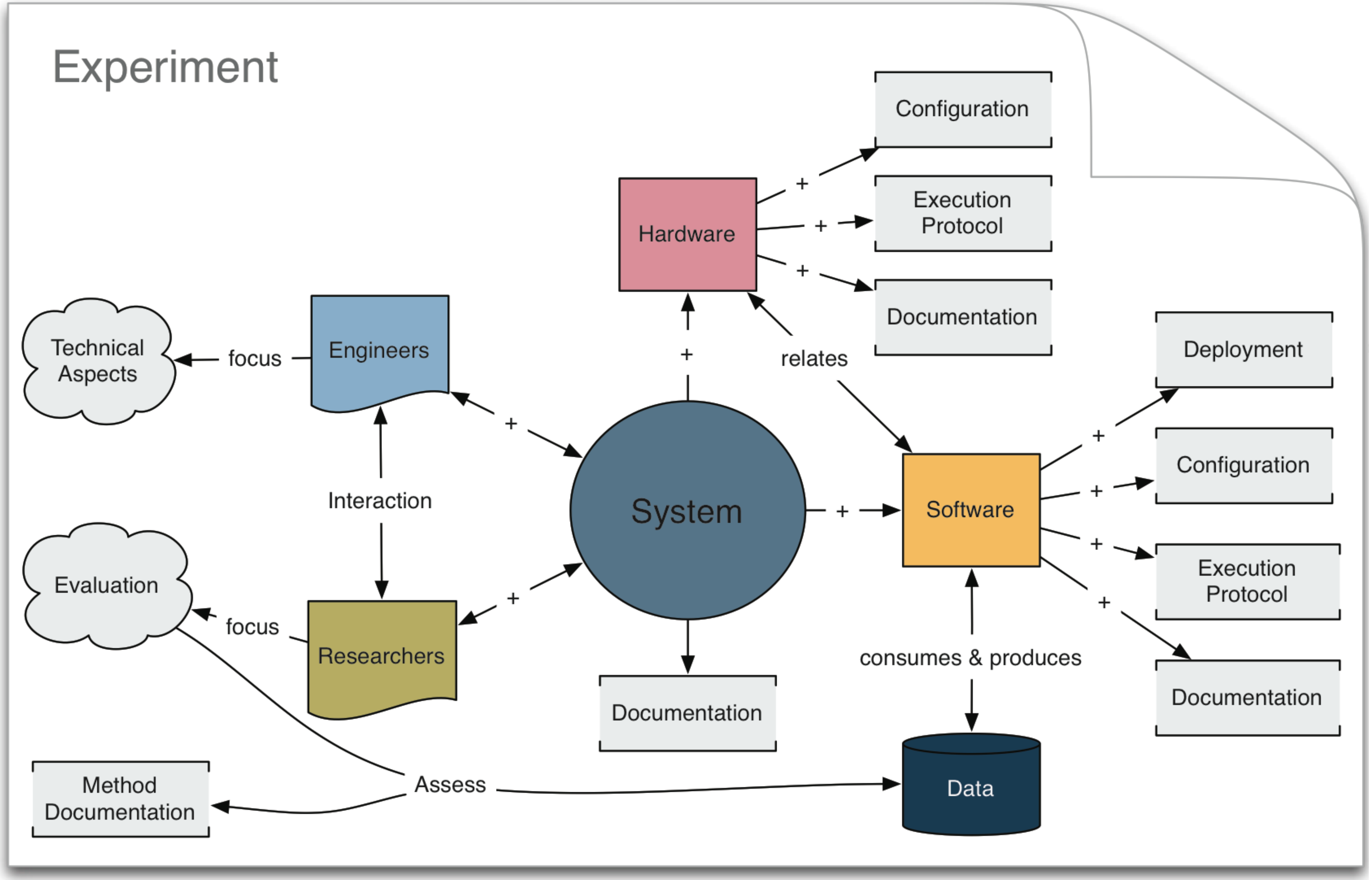
» Numerous studies have been performed over the past 15 years, but one of the hallmarks of science has yet to be achieved: results at present are hardly ever reproducible by other research groups. « [2]

Methodological issues that prevent reproducibility of robotic system experiments. [3]

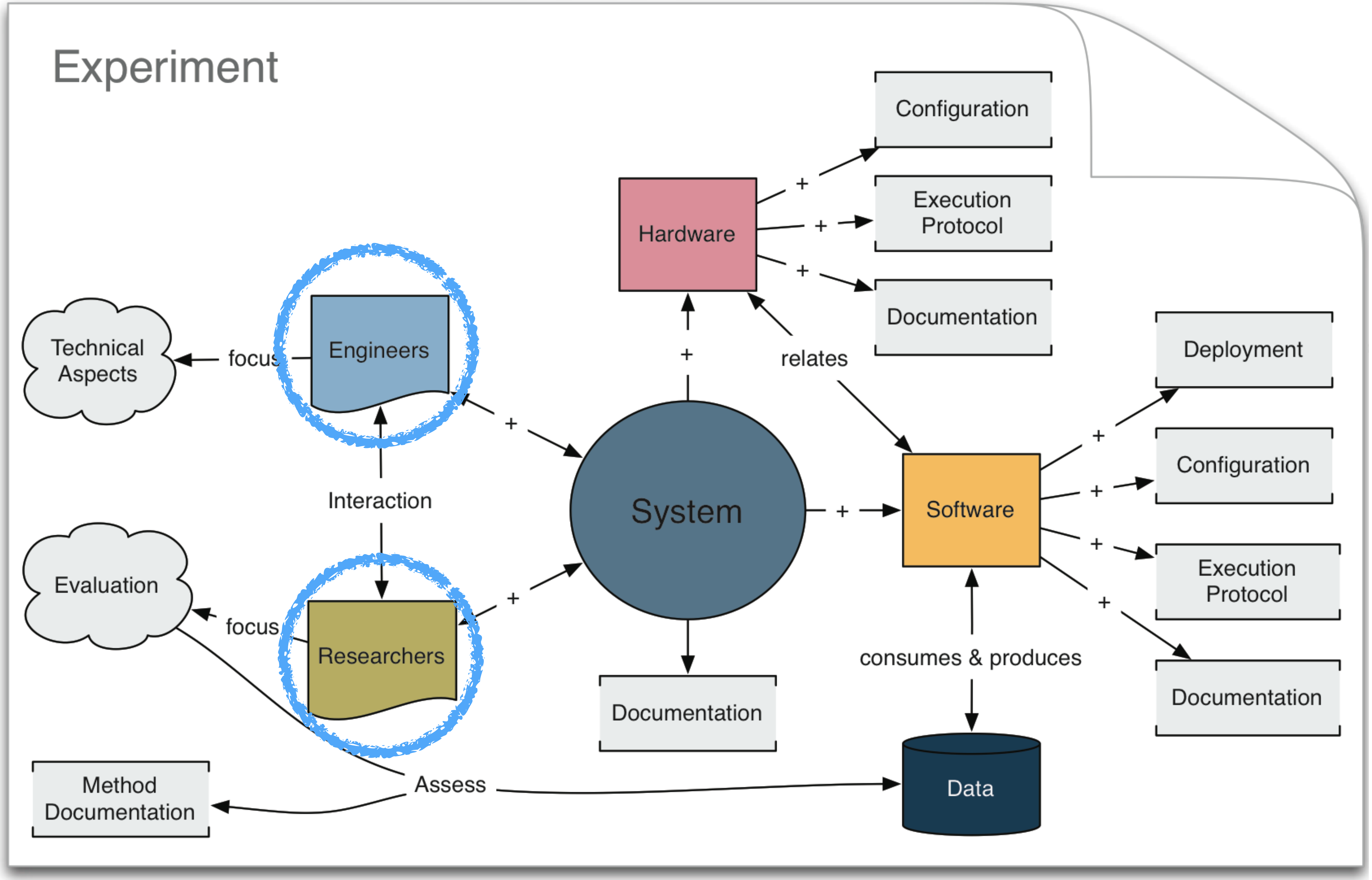
This includes the frequently neglected impact on experiments caused by the relationship between **individual** components and the **whole** system — in component-based systems for instance.

Question: how do publications need to be written and published in order to **improve reproducibility**?

Systems in Experimental Robotics | **Aspect Overview**



Systems in Experimental Robotics | **Overview**



Information retrieval and aggregation: artifacts are often distributed over different locations, thus already the discovery, identification and aggregation of all required artifacts is difficult.

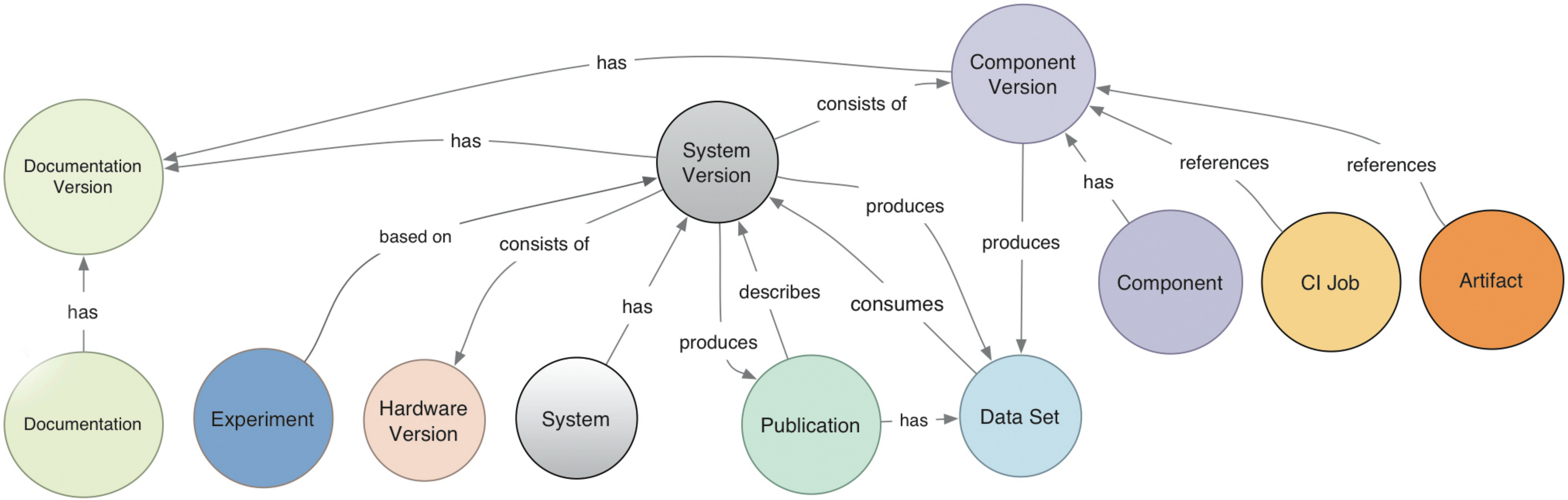
Semantic relationships: which specific versions (master/v. 133.7) of software components were in use for a particular study?

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.

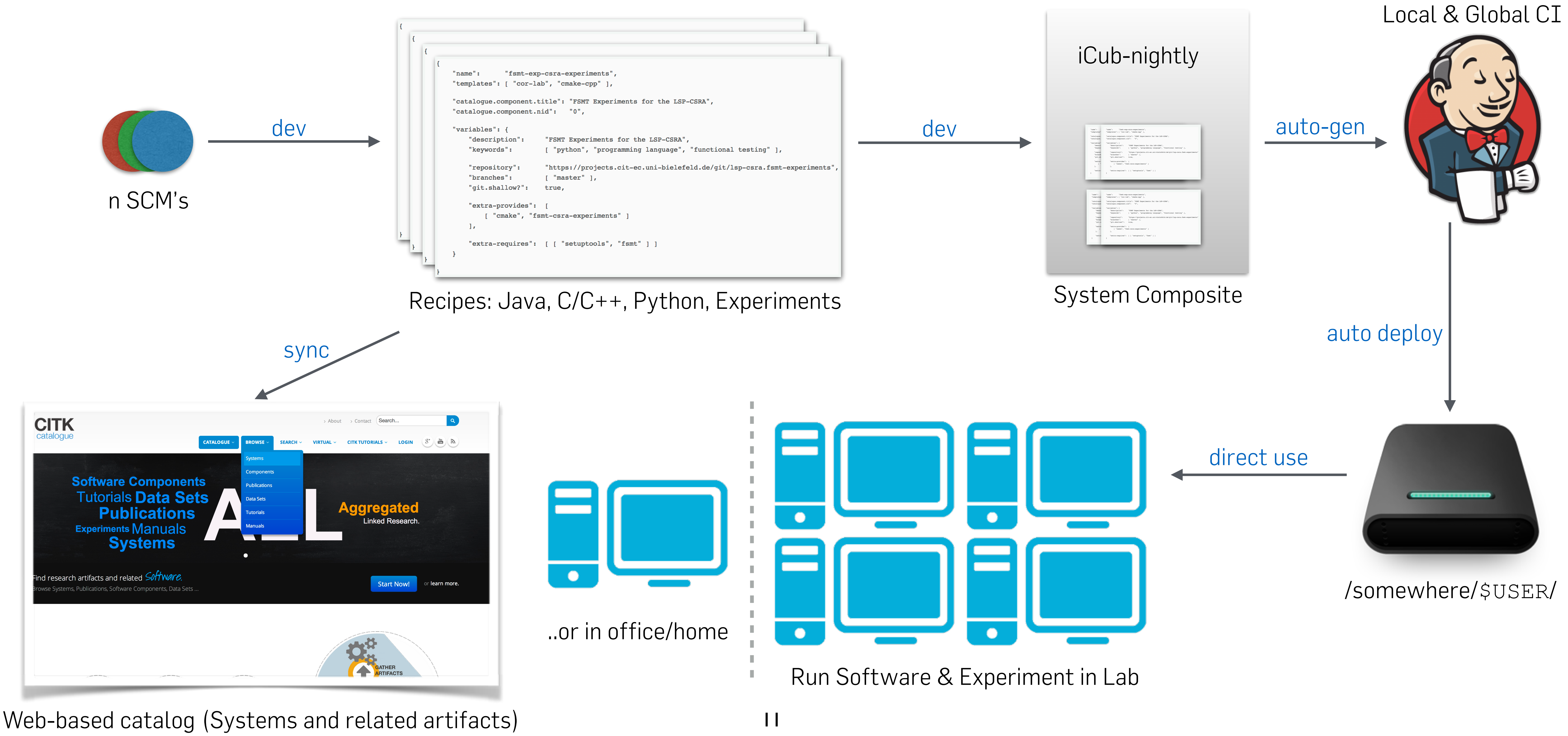
Experiment testing, execution and evaluation: Advanced robotics experiments require significant efforts spent on system development, integration testing, execution, evaluation and preservation of results. This is particular costly if many of these tasks are carried out manually. Crucial run time parameters 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**.

Systems in Experimental Robotics | **System Model**



Systems in Experimental Robotics | **Technical Realization Overview**



Systems in Experimental Robotics | **Technical Realization Recipes**


```
{

    "name":      "yarp",
    "templates": [ "cor-lab", "cmake-cpp" ],

    "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", "cpp" ],

        "repository":      "https://github.com/robotology/yarp.git",
        "branches":        [ "master", "v2.3.63" ],
        "git.shallow?":    false,

        "extra-provides":  [
            [ "cmake", "yarp", "2.3.63" ],
            [ "cmake", "YARP", "2.3.63" ]
        ],

        "cmake.options":   [ "CMAKE_BUILD_TYPE=Release",
                             "CREATE_GUIS=ON",
                             "CREATE_LIB_MATH=ON",
                             "CREATE_OPTIONAL_CARRIERS=ON",
                             "CREATE_SHARED_LIBRARY=ON",
                             "ENABLE_yarpcar_bayer_carrier=ON",
                             "@{next-value|[]}" ]

    }

}
```

Systems in Experimental Robotics | **Technical Realization Recipe (Experiment)**


```json
{
    "name":      "fsmt-exp-icub-nightly",
    "templates": [ "cor-lab", "cmake-cpp" ],

    "catalogue.component.title": "FSMT Experiment iCub Ball Tracking",
    "catalogue.component.nid": "1276",

    "variables": {
        "description":           "Replay recorded data set and make the icub track a red ball",
        "keywords":              [ "fsmt", "functional testing", "iCub" ],

        "repository":            "https://openresearch.cit-ec.de/git/fsmt.experiments.git",
        "branches":              [ "master" ],
        "sub-directory":         "icub-nightly",

        "extra-requires":        [ [ "setuptools", "fsmt" ] ],

        "redmine.instance":      "https://openresearch.cit-ec.de/",
        "redmine-project":       "fsmt"
    }
}
```

Systems in Experimental Robotics | **Technical Realization Experiment Protocol**

Local & Global CI

..or in office/home

Lab

Web-based catalog (Systems and related artifacts)

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

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 oder 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

Find CITK link in publication

CITK catalogue

› About › Contact Search...

CATALOGUE BROWSE SEARCH VIRTUAL CITK TUTORIALS FLIER'S CONTRIBUTIONS

Home / Systems / iCub Ball Tracking / Version /

iCub Ball Tracking-nightly

View | Edit | Track | Devel

Version: nightly
Acronym: icub-btracking
Version of System: iCub Ball Tracking
Videos: iCub Ball Tracking Demo Video
Description: In this particular system version, iCub related tools (yarpview, iCubGui and dataSetPlayer) are used replay a pre-recorded sequence as provided in the iCub Wiki in order to visualize the original state of the robot. Feel free to **1) replicate** the system and **2) execute** the linked experiment to verify our results. Please see Linked Artifacts section. For a first glimpse, you may also check the linked video above.

System Replication

For now, we have tested the replication with Ubuntu 14.04 (Trusty). Please install the following packages first.

In the Bielefeld University/TechFak environment you don't need to install these packages

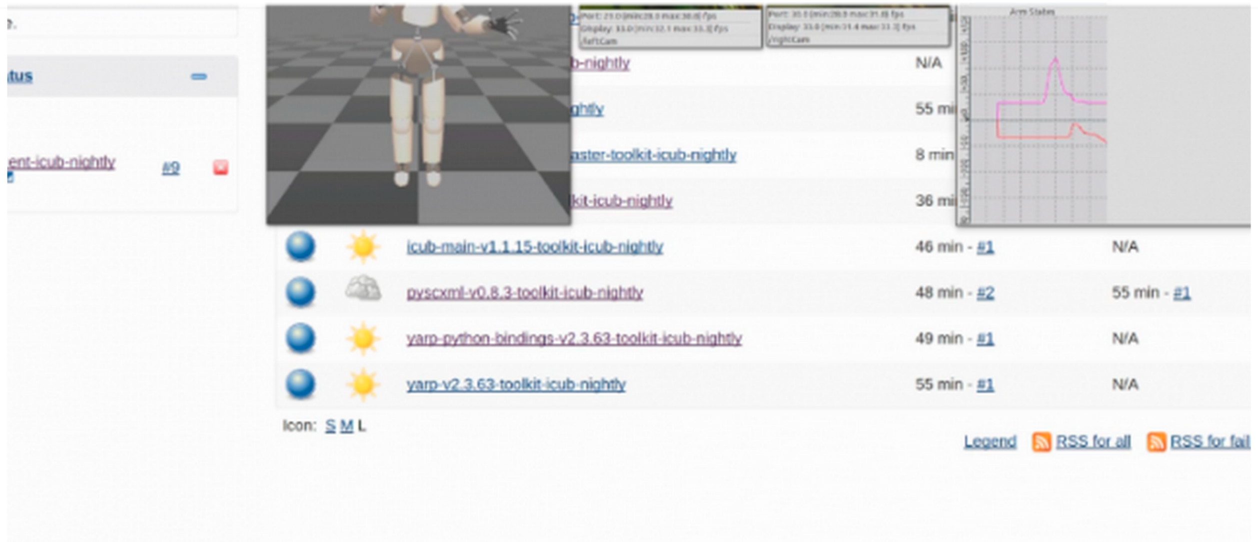
Ubuntu 14.04 LTS

```
sudo apt-get install libace-dev libgsl0-dev libc6 python libncurses5-dev libgtkmm-2.4-dev libglademm-2.4-dev libqt4-dev \
libqt4-opengl-dev libcv-dev libhighgui-dev libcvaux-dev libsdl1.2-dev  git gfortran freeglut3-dev cmake libxmu-dev libode1 \
libode-dev libopencv-dev libblas-dev python-dev openjdk-7-jdk swig gnuplot eog unzip
```

Now, please use our distribution tool chain as explained in the tutorials section Bootstrapping and Installing. You may skip the MAVEN part as it is not required for this system version. Read and execute these instructions carefully. You will need to bootstrap the **icub-nightly.distribution.** If you changed your `$prefix`, please keep that in mind.

```
$prefix/jenkins/job-configurator --on-error=continue -b http://localhost:8080 -d /$prefix/dist/distributions/icub-nightly.distribution -t '$prefix/
dist/templates/*.template'
```

Visit system version on catalog website



GENERAL INFORMATION

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
PySCXML-v0.8.3
Linked Distribution Recipe:
icub-nightly
Linked Experiments:
iCub Ball Tracking-nightly

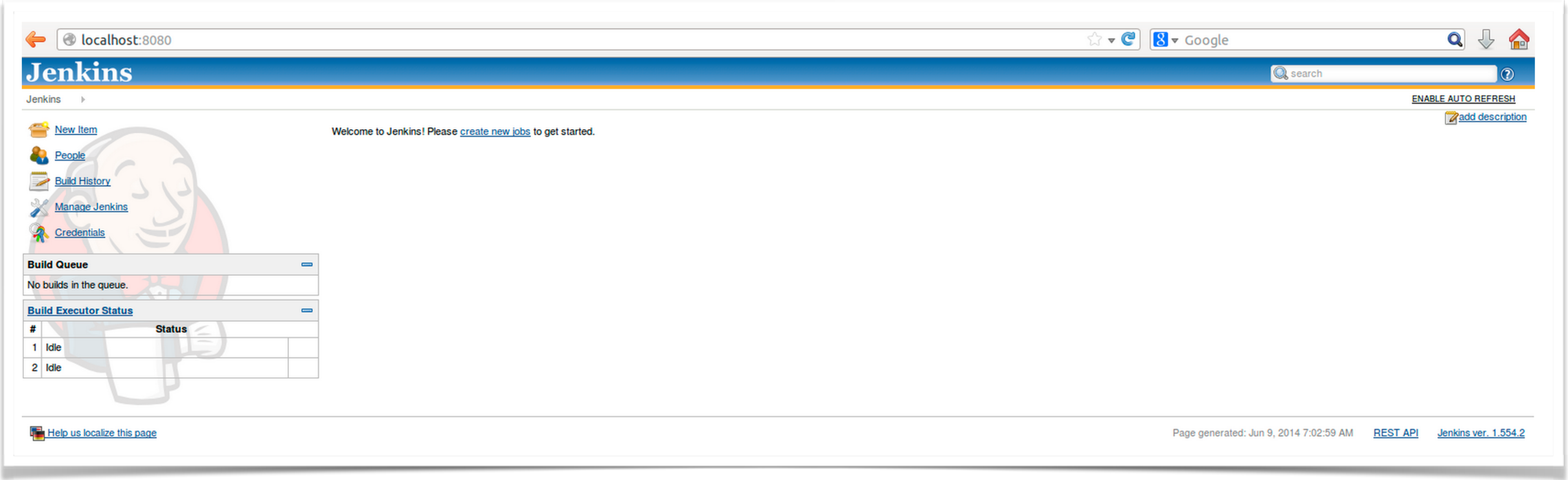
Browse linked artifacts: Publications, Component Versions, Data Sets, etc...

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

Download pre-packaged CI Server

```
cd $prefix/jenkins
./start_jenkins
```

Extract & simply start it



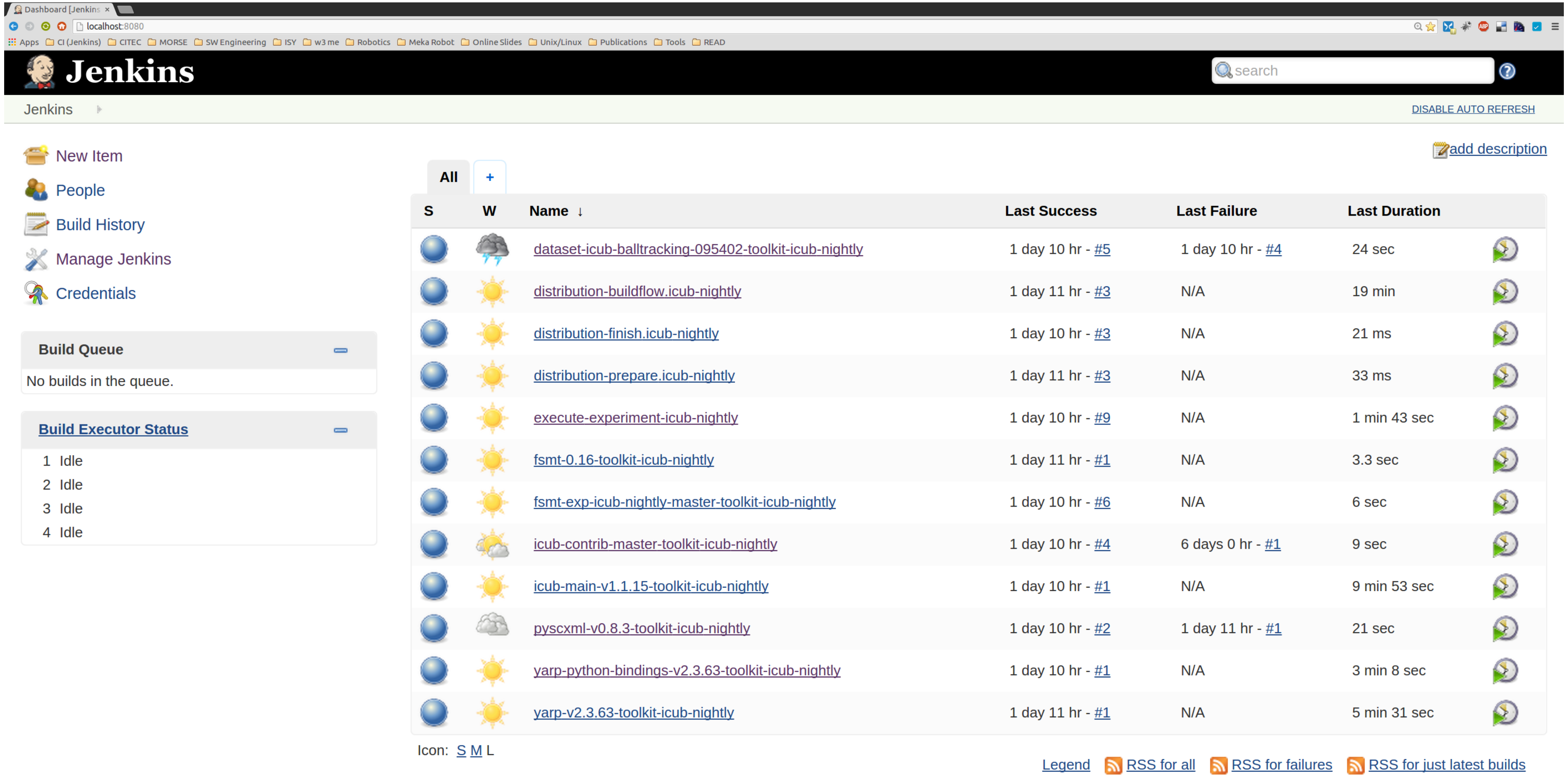
Find ready to use local CI Server

```
cd $prefix
mkdir dist
cd dist
git clone https://opensource.cit-ec.de/git/citk .
```

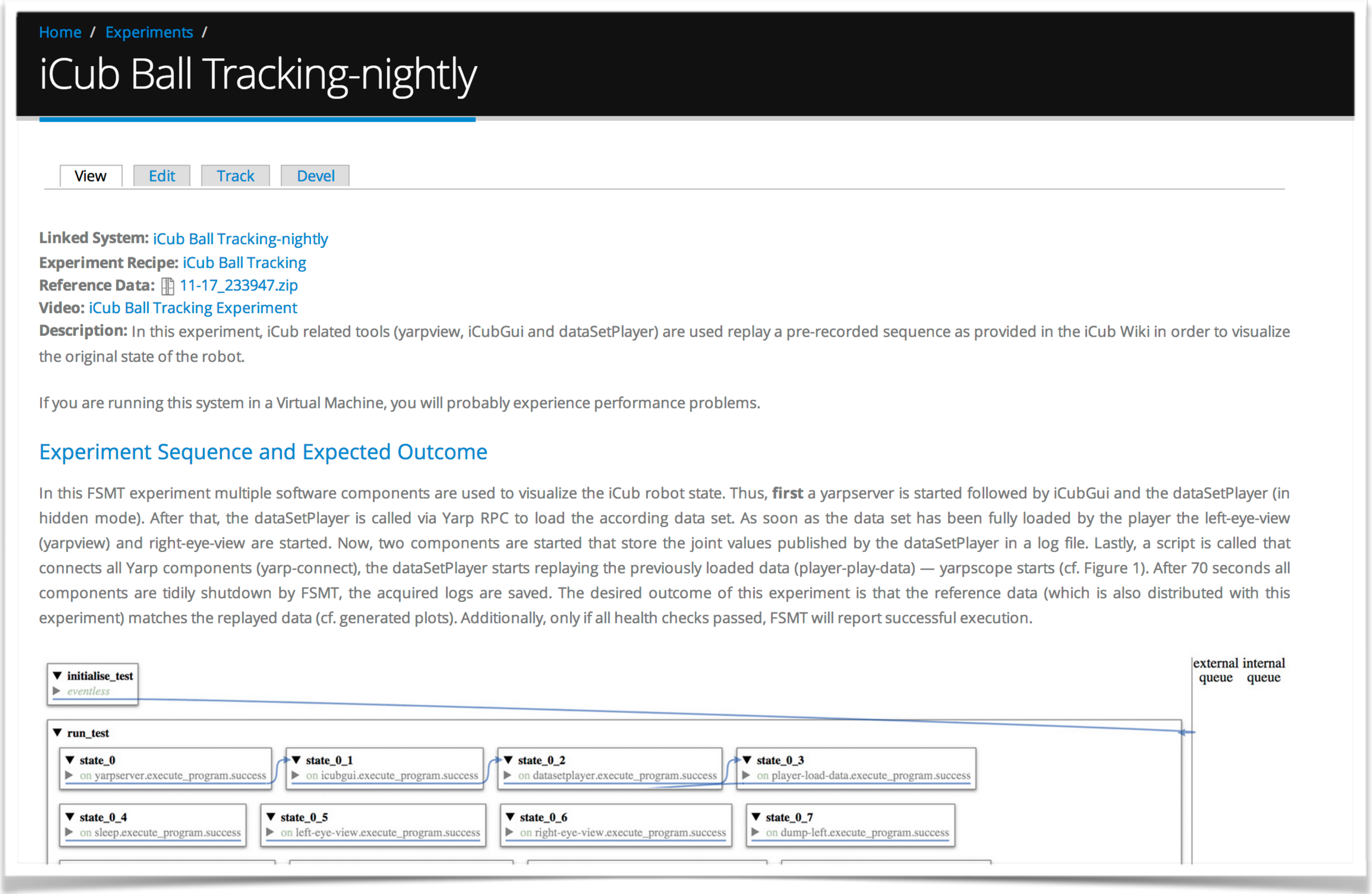
Clone CITK repository

```
/tmp/jenkins/job-configurator --on-error=continue -b http://localhost:8080 -d /tmp/dist/distributions/DESIRED_DISTRIBUTION.distribution -t '/tmp/di
st/templates/*.template'
```

Invoke job-configurator with desired distribution



Start build-flow job, the rest is orchestrated automatically, system is fully built (repeatable via single click)



Home / Experiments /

iCub Ball Tracking-nightly

View | Edit | Track | Devel

Linked System: iCub Ball Tracking-nightly
Experiment Recipe: iCub Ball Tracking
Reference Data: 11-17_233947.zip
Video: iCub Ball Tracking Experiment
Description: In this experiment, iCub related tools (yarpview, iCubGui and dataSetPlayer) are used replay a pre-recorded sequence as provided in the iCub Wiki in order to visualize the original state of the robot.

If you are running this system in a Virtual Machine, you will probably experience performance problems.

Experiment Sequence and Expected Outcome

In this FSMT experiment multiple software components are used to visualize the iCub robot state. Thus, **first** a yarpserver is started followed by iCubGui and the dataSetPlayer (in hidden mode). After that, the dataSetPlayer is called via Yarp RPC to load the according data set. As soon as the data set has been fully loaded by the player the left-eye-view (yarpview) and right-eye-view are started. Now, two components are started that store the joint values published by the dataSetPlayer in a log file. Lastly, a script is called that connects all Yarp components (yarp-connect), the dataSetPlayer starts replaying the previously loaded data (player-play-data) — yarpscope starts (cf. Figure 1). After 70 seconds all components are tidily shutdown by FSMT, the acquired logs are saved. The desired outcome of this experiment is that the reference data (which is also distributed with this experiment) matches the replayed data (cf. generated plots). Additionally, only if all health checks passed, FSMT will report successful execution.

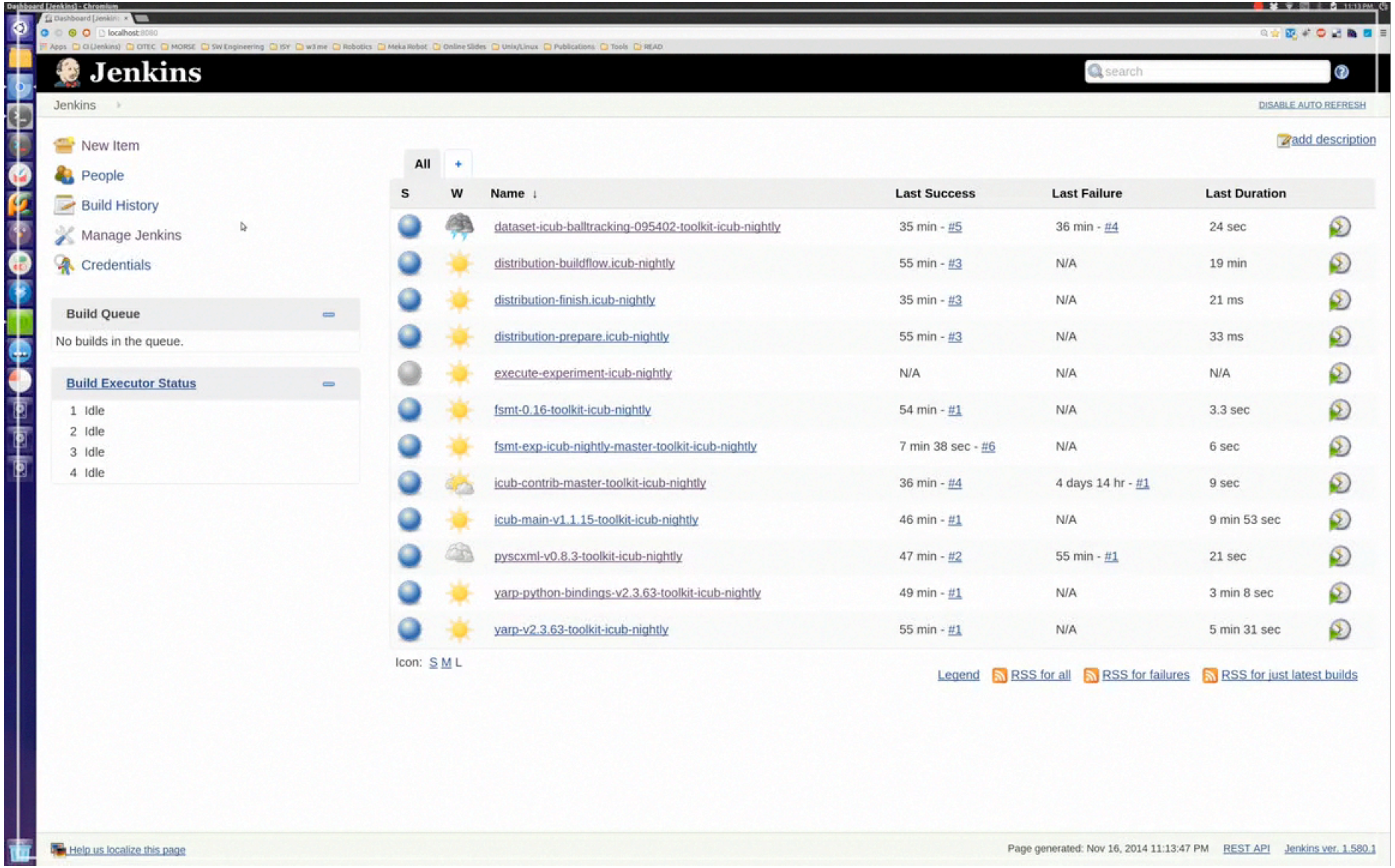
Visit linked experiment in the web catalog and review reference data

```
export prefix=/vol/icub-nightly/
export PATH=$prefix/bin/:$PATH
export PYTHONPATH=$PYTHONPATH:$prefix/lib/python2.7/site-packages/
fsmt $prefix/etc/fsmt-experiments/icub-nightly/icub-nightly-balltracking.scxml
```

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

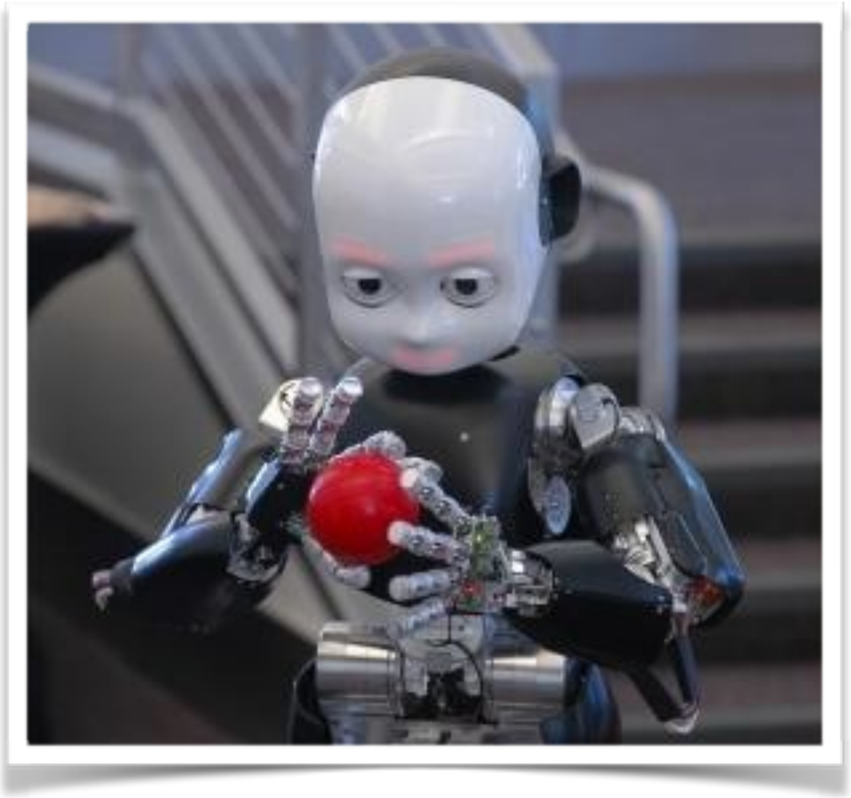
Support for more than 2 Linux flavors, Mac OSX, probably Windows too

… 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!

ISTITUTO ITALIANO
DI TECNOLOGIA



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

[1] A. Bonarini et al. RAWSEEDS: Robotics advancement through web- publishing of sensorial and elaborated extensive data sets. In IROS'06 Work- shop on Benchmarks in Robotics Research, volume 6, 2006.
[2] K. Dautenhahn. Methodology and themes of human-robot interaction: a growing research field. Internatio- nal Journal of Advanced Robotic Systems, 2007.
[3] F. Bonsignorio, J. Hallam, and A. del Pobil. Defining the requisites of a replicable robotics experiment. In RSS2009 Workshop on Good Experimental Methodologies in Robotics, 2009.

HANDS ON SESSION — Let’s integrate?