Presentation and organization of the RSCPS lab

Andrea Ceccarelli
andrea.ceccarelli@unifi.it

My introduction

Research interests: design and assess fault tolerant and secure systems

- system monitoring, authentication, testing, …
- software-intensive sytems, IoT, service-oriented architecture

Teachings AA2017-2018

- Operating Systems, Quality & Certification, Distributed Real-Time Cyber Physical Systems

Scope of the lab

Hands-on experience with a **cyber-physical distributed system (of systems)** where constituent systems

Interacts and
coordinates
using cyber and stigmergic channels
in order to achieve a shared goal

Practise with the life-cycle phases:

- Requirements definition
- (Model-based) design
- Implementation
- Deployment

(no assessment)

"In theory, theory and practice are the same. **In practice, they are not**."

Required competences and compentences to be acquired

Design

- Sytem-of-System engineering (available)
- Blockly and Blockly4SoS (missing)

Programming

- C programming (available)

Robotics

- Kilobots simulation (missing)
- Kilobots deployment (missing)

Target application: Strega-comanda-color (Witch Says Colors)

The players must be at least 3.

With a counting out rhyme, a "WITCH" is selected.

The Witch calls out a colour; all the players must go and touch a "thing" of that colour: it can be clothes, objects, or anything in the nature and environment.

The game ends when the witch touches a player before he/she can touch the wanted
colour.

This player will be the witch for the next round of the game.

Target application: Strega-comanda-color (Witch Says Colors)

- Kilobots have led of different colors, each kilobot sets a color
- One Kilobot is nominated the WITCH
  - Chooses a led color
- The kilobot with the nominated colored led runs away (runner)
- All the others try to catch (catchers)
- The game ends when the running Kilobot is captured
- Work in team, implement both runner and catchers software

We will need to decide on some requirements engineering questions:

- Initial positioning?
- Communication protocol?
- How to define "capture"?
- What should be communicated while moving?

Let's have a look at the lifecycle we will explore during lab

Requirements definition

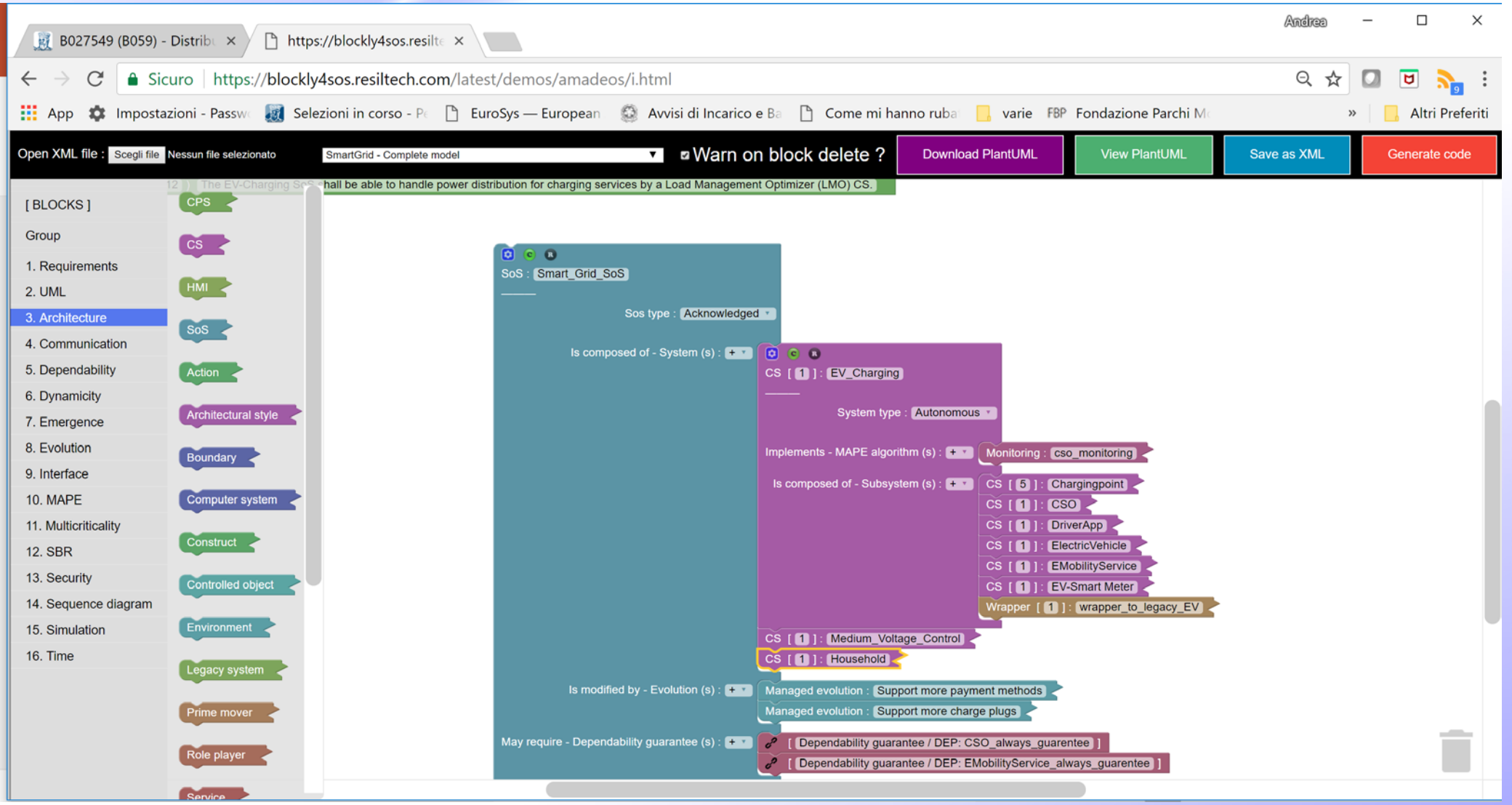
We apply concepts of Systems of Systems.

Requirements definition following viewpoints (partially in class):

- Architecture
- Interfaces (RUI)
- Dynamicity
- Evolution
- Emergence
- Time
- Security
- Dependability

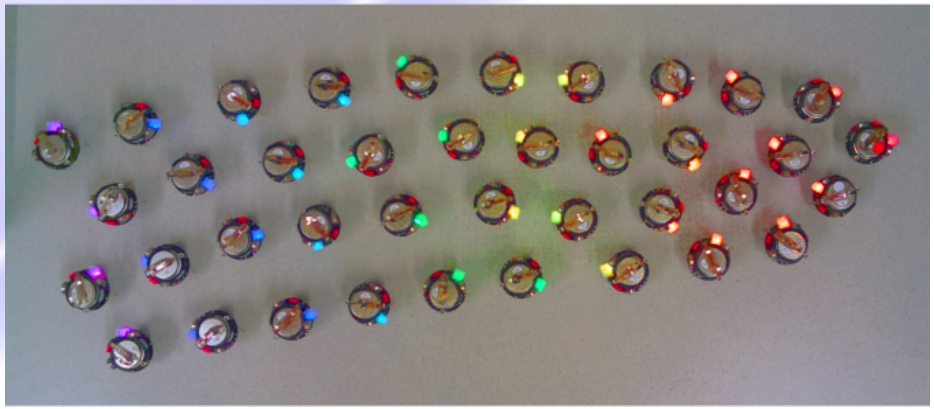
(governance, any other?)

Model-based design

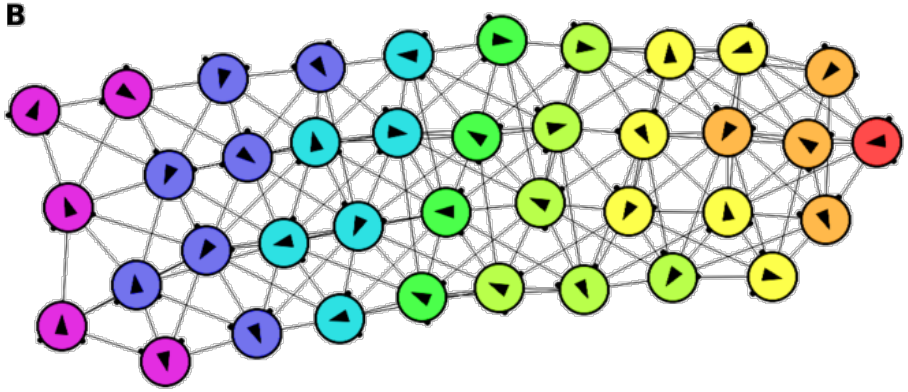


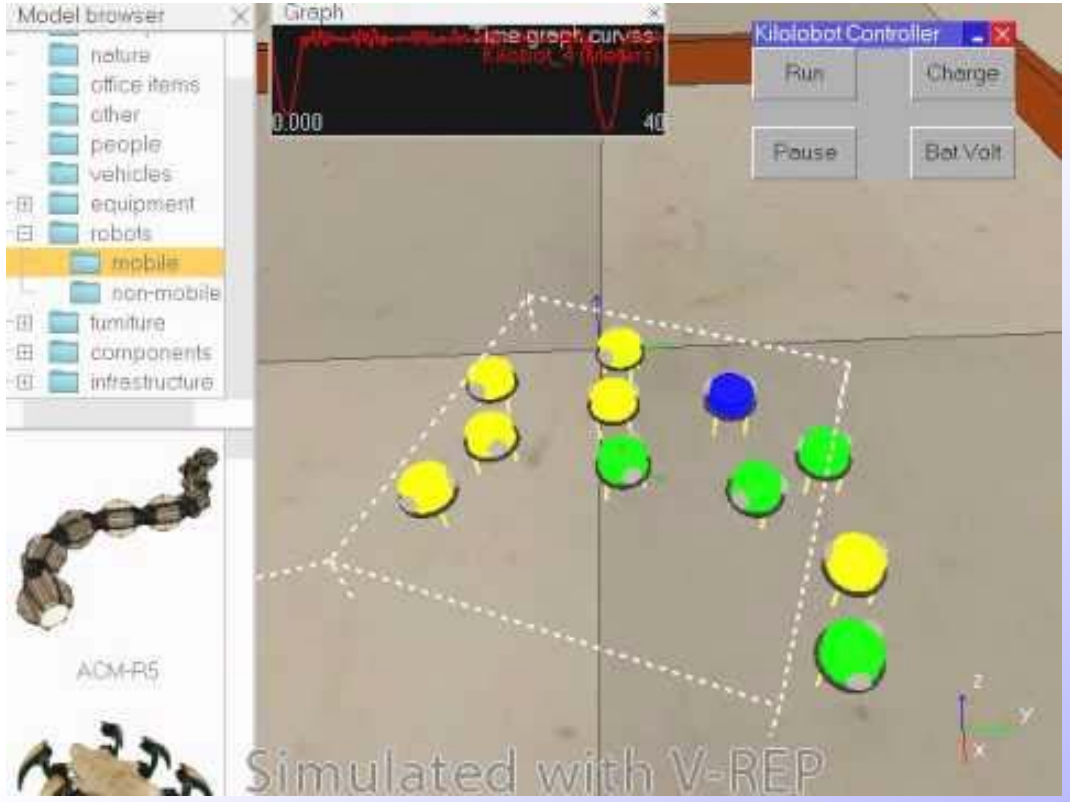
Simulation and prototyping

A



B





DISTRIBUTED REAL TIME CYBER PHYSICAL SYSTEMS

Coding and deploy

Files

prova.c

untitled.c

Save | Compile | Editor Options

```
1  #include <kilolib.h>
2  #define DEBUG
3  #include <debug.h>
4
5
6  message_t message;
7  // Flag to keep track of message transmission.
8  int message_sent = 0;
9  // Flag to keep track of new messages.
10 int new_message = 0;
11 int distance = 0;
12 int Mess0 = 0;
13 int Mess1 = 0;
14 static int TestLED=0;
15
16 void setup()
17 {
18
19     // Initialize message:
```

More in details: what is a KILOBOT

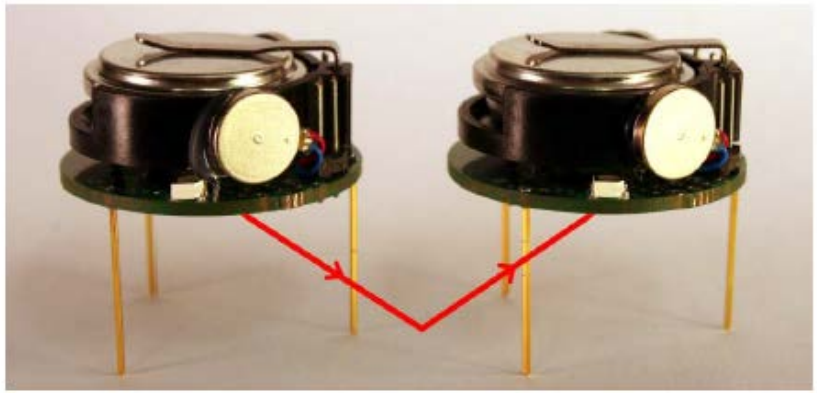
Low cost robots designed at Harvard University's Self-Organizing Systems Research Lab

- http://www.eecs.harvard.edu/ssr.

➢ Designed to test collective algorithms (swarm robotics)

They include:

- differential drive locomotion
- on-board computation power
- neighbor-to-neighbor communication
- neighbor-to neighbor distance sensing
- ambient light sensing



Some technical specs

The main specifications of the Kilobot robot are listed below:

- Processor : **ATmega 328p (8bit @ 8MHz)**
- Memory : **32 KB Flash**
- 1KB EEPROM
- Battery/ autonomy : Rechargeable Li-Ion 3.7V / 3-10 hours continuously, 3 months in sleep mode
- Communication : **IR (up to 7cm**, up to 32kb/s and 1kbyte/s with 25 robots),
- Sensing: **1 IR and 1 light intensity**
- Movement : forward, rotation **(1cm/s , 45deg/s)**
- Light : one RGB led
- Dimensions : diameter: 33 mm, height 34 mm

Some words on the equipment and the setup (room 1/20 DiMaI)

10 Kilobot robots

An overhead controller

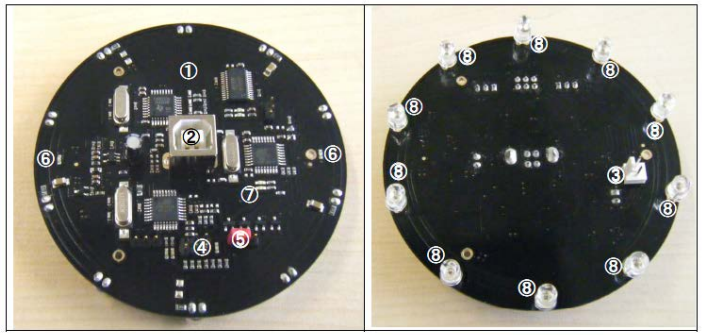


Figure 3.2: Overhead controller overview

A charger

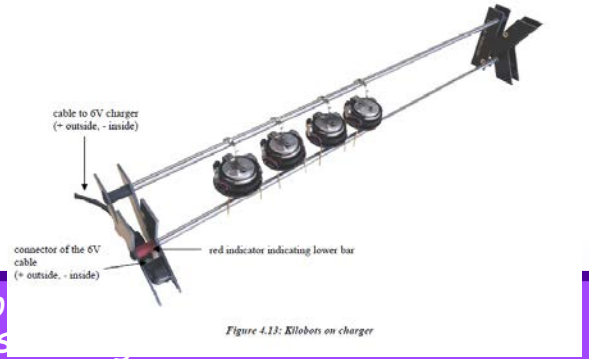
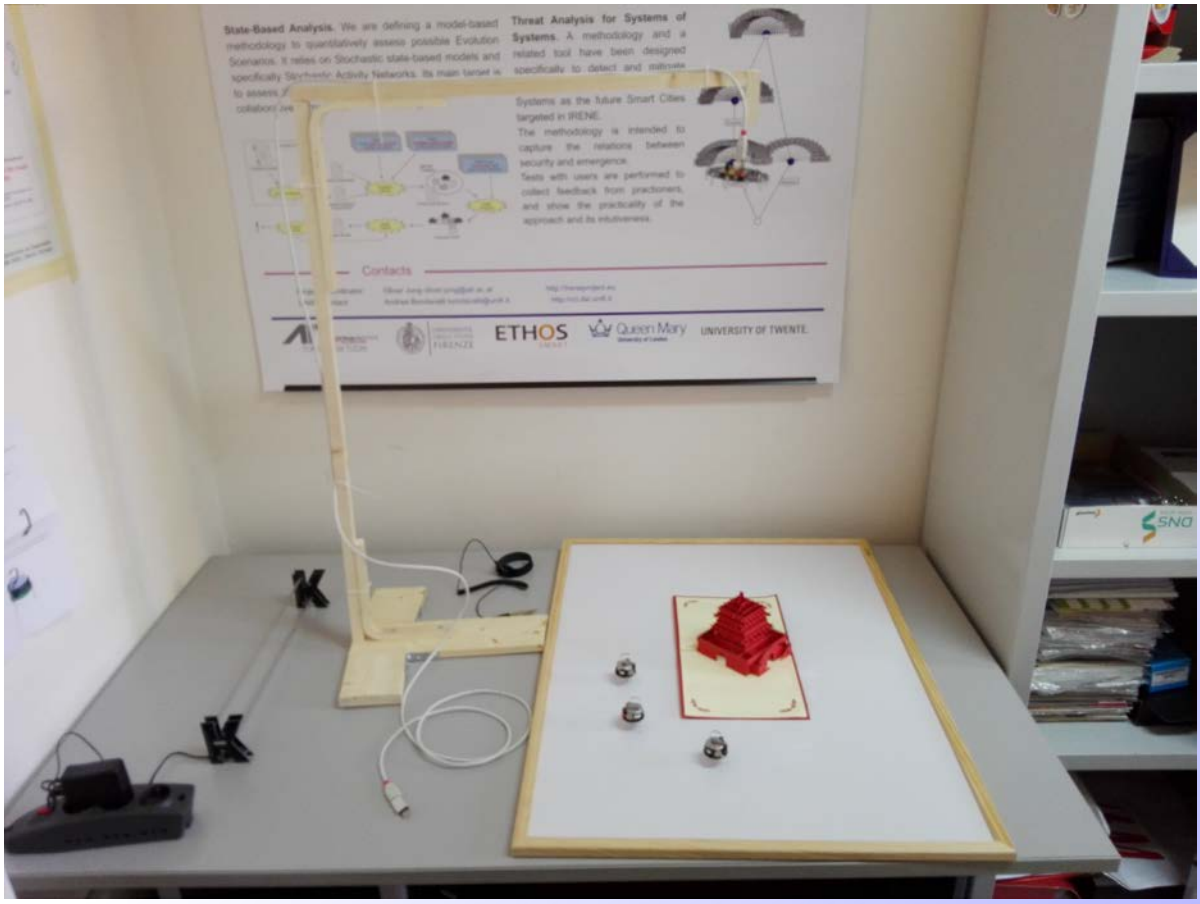


Figure 4.13: Kilobots on charger



Some technical difficulties in moving the setup outside the Department

Lab structure and organization

Main facts – time and logistic

Approx. 18 hours

Rooms (when not specified, lecture is in the usual class)

19-11-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	09:30 - 11:30
20-11-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	09:30 - 11:30
21-11-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	10:30 - 13:30
26-11-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	09:30 - 11:30
27-11-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	09:30 - 11:30
03-12-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	09:30 - 11:30
04-12-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	09:30 - 11:30
05-12-2018	C.Didat.Morgagni(1°piano) Aula Informatica 110	10:30 - 13:30

Possibly, *bring your own laptop*

- How many laptops are available?

Main facts - lab execution

First part of the lab (6-8 H): learn the enabling technologies

- will be explained by the Teacher
- some hands-on → active participation is recommended

Second part of the lab (approx. 10-12 H): apply the technologies

- The activites are started in class during the lesson, in groups 1-3 students (please be interactive)
  - Scrum sprints & hackaton approaches
- Then the students are asked to complete the work offline

Tentative structure of the labs

Approx 18 Hours

- 2 H: Blockly4SoS and examples/practise
- 2 H: Kilombo (Kilobots simulator) usage and examples/practise
- 3 H: Kilobots usage and examples/practise
- 2 H: Requirements definition LAB- students main actors
- 2 H: Modeling in Blockly4SoS LAB- students main actors
- 3 H: Kilombo implementation LAB- students main actors
- 4 H: Kilobot implementation LAB- students main actors

(possible) contest: date not set.

Offline: contact teacher to execute on the real Kilobots (to test software on the real thing)

Exam: submission and evaluation

Main facts -The work requested

Students are asked to:

- Define requirements, considering the different CPSoS viewpoints, for the Kilobot-based Witch Says Color. The starting points are the requirements defined in class, but **modifications and additions are allowed when not necessary**.
- Define the architecture (and sequence diagrams, at least for a representative selection) for the Kilobot-based Witch Says Color, using the Blockly4SoS tool. Consider the possibility to explore various viewpoints, and remember the possibility to trace requirements.
- Implement the defined architecture for the Kilombo simulator.
- Implement the defined architecture for the Kilobots.

What and how to deliver

Each group is asked to submit the following work products:

- A pdf report containing:
  - the list of requirements
  - the design: an overview of the Blockly4SoS model – only main facts!
  - the implementation: the realization of the design – only main facts!
  - discussion on the **simulated** execution – only main facts!

- A zip or tar file containing
  - The Blockly4SoS xml file
  - The code for the KILOMBO simulator
  - The code for the Kilobot execution

Main facts - evaluation

The work will be evaluated thorough the inspections of work products and an interview on the work products.

Evaluation means and criteria. Students are evaluated on the:

- development of the lab exercise (work products delivered)
- interview on all the work done to develop the lab exercise
- during the interview, students may be asked to operate with the developed work products, including running or modifying the software and/or running Kilobots.

Consequently, the score is based on the:

- quality of the work products delivered
- quality of the interview
- (possibly) ability in running the KILOBOT software during the exam

How and when

How to deliver the lab exercise

- Via mail to andrea.ceccarelli@unifi.it
- If no ack is received after 5 working days, WRITE AGAIN

When to deliver

- Deadlines are: 14 days (two weeks) before exams date
  - When you deliver the lab exercise, we can arrange a date for the interview, but it must be a minimum of 2 weeks from such delivery date.
- You are welcome to deliver any time of the year → discussion can take place anytime

How to arrange a date for the oral discussion

- By mail, preferably the whole group togheter

List of software resources needed

- Any **Internet browser** is OK (Chrome, Firefox, Internet Explorer, …)
- Any **editor** for C programming and the **GCC compiler**
- **Kilombo** (runs on Linux or OSX) https://github.com/JIC-CSB/kilombo
  - see the README for installation instructions
  - Detailed installation instructions, that may be necessary for some UNIX distributions, are at http://jic-csb.github.io/kilombo/
- **KiloGUI:** This GUI is for using the overhead controller to start the kilobots, upload new programs, calibrate the robots, etc. Installation is very simple and supported for multiple platforms. Usage instructions are below. Go to this link to download the GUI for the Linux OS: https://www.kilobotics.com/download
  - The kilobots are programmed using a C-like language (AVR C). See the Kilobot Library API (http://wwww.kilobotics.com/docs/index.html) and also see the Labs (http://www.kilobotics.com/labs) tab for a tutorial on how to start programming the robots.
- **Editor and Compiler:** The Kilobotics Web-based Editor (http://www.kilobotics.com/editor) allows anyone to write and compile programs for the kilobot.
  - Further information on using the KiloEdit web-based editor:
    A **Dropbox account** is necessary. Dropbox software must be locally installed. The first time the Editor is open, it will ask to sign into Dropbox.

References by topics

Design

- Blockly and Blockly4SoS
  - https://blockly4sos.resiltech.com/
  - https://blockly4sos.resiltech.com/downloads/demo/blockly4sos_demo-v1.2.avi
  - https://blockly4sos.resiltech.com/user-guide.pdf
  - Book "Cyber-Physical Systems of Systems" (open access), chapters
    - Basic Concepts on Systems of Systems
    - (AMADEOS SysML Profile for SoS Conceptual Modeling)
    - AMADEOS Framework and Supporting Tools
    - Case Study Definition and Implementation

https://link.springer.com/book/10.1007/978-3-319-47590-5

References by topics

Robotics

- Kilobots and kilobots programming
  - Manuals available on course web-site
  - www.kilobotics.com
    - Editor: https://www.kilobotics.com/editor
    - Tutorials: https://www.kilobotics.com/labs
  - https://www.k-team.com/
  - https://www.youtube.com/playlist?list=PLC7119C2D50BEA077
  - Rubenstein, Michael, Christian Ahler, and Radhika Nagpal. "Kilobot: A low cost scalable robot system for collective behaviors." *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012
- For simulation
  - KILOMBO
    - JANSSON, Fredrik, et al. Kilombo: a Kilobot simulator to enable effective research in swarm robotics. *arXiv preprint arXiv:1511.04285*, 2015.
    - http://jic-csb.github.io/kilombo/