RobMoSys
composable models and software for robotics systems

Project Coordinator: Sara Tucci
CEA LIST, Paris-Saclay, France
HORIZON 2020 project nº 732410
Innovation Action on systems development technology
www.robmosys.eu
The project ID

- **RobMoSys**: Composable Models and Software for Robotic Systems
- In response to H2020 Project – ICT-26- TOPIC : System abilities, development and pilot installations
- SubTopic c: **Innovation Action** on systems development technology.
  The “**System development tools**” sub-call

- **Start Date** 01/01/2017
- **End Date** 31/12/2020
- **Duration** 4 Years
- **Budget** 8M, where 4 M for Open-Calls
- **Web Site** [http://robmosys.eu/](http://robmosys.eu/)
**AMBITION**

RobMoSys will coordinate the whole community’s best and consorted efforts to realize a step-change towards an industry-grade software development European ecosystem

- open
- sustainable
- industrial quality

RobMoSys part of the effort on Digital Industrial Platforms for Robotics
Model-driven engineering as key enabler for

- complex software and system integration
- high-quality standards compliance

Expected results

- repository of formal models for robotics
- tooling for correct-by-construction composition and code generation
- open-source software and tools “track”

Community Involvement

- Workshops Series for fast feedback from the community: 1° today at ERF 16.15 – 17.45!
- Open Calls (better tools, models and software)
IMPACT

Professionalization of software development in robotics

Improvement of product and service quality at reduced development costs

The further commoditization of basic robotics software, models and advanced modeling tools

Grow the potential market of new-generation robotics systems
Models for Composition in an Ecosystem with Separation of Roles

22 March 2017
European Robotics Forum (ERF), Edinburgh
System Composition in an Ecosystem

- RobMoSys adopts a **composition-oriented** approach to **system integration** that manages, maintains and assures **system-level properties**, while preserving **modularity** and independence of **existing robotics platforms** and code bases, yet can build on top of them.

- Towards an open and sustainable European robotics software **ecosystem** based on models and supporting **separation of roles**

- Apply **model-driven** techniques

- Manage **non-functional properties**

- From integration-centric to composition-oriented approaches

We operationalize architectural patterns and composition such that properties of system-of-systems become known in order to build trust in the system under development.
Imagine you have **developed software** to localize a robot in the environment and you are interested in **making it available** in robotics.

You are
- a **SME**, specialized in a certain domain
- e.g. a **component supplier** for robot navigation

You want to
- express your offer with pivotal features such that others can find your component (yellow pages)
- ensure that others can use your component (composability + compositionality)
- explicate non-functional properties of your component and define its variation points
Imagine you as an integrator are willing to develop an application which **needs a localization** module and you are interested in **integrating the third-party localization** software on your intralogistics mobile platform.

You are

- a SME that wants to **access robotics technology** and that wants to **build a robot** application

You want

- to **select components** from the market matching your expressed needs
- your application to be **correct by construction**: you expect that building blocks seamlessly fit together
- to view components as **grey-boxes** and use them “as-is”: adjust only at explicated variation points within modeled boundaries, do not modify source code
### Platform Economy

<table>
<thead>
<tr>
<th>Levels</th>
<th>Concerns</th>
<th>Computation</th>
<th>Communication</th>
<th>Coordination</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission</td>
<td>serve as butler</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>deliver coffee</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skill</td>
<td>grasp object</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>with constraint</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>move manipulator</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>IK solver</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execution Container</td>
<td>activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating System / Middleware</td>
<td>pthread, socket</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>manipulator, laser scanner, CPU</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**The Challenge:**

- Adhere to separation of concerns while at the same time, package different concerns into structures such that these fit the views of the different roles.
- Manage interfaces between different roles and different levels of abstractions.
Ecosystem View and Separation of Roles

- RobMoSys enables the composition of robotics applications with managed, assured and maintained system-level properties via model-driven techniques.

- RobMoSys enables *communication of design intent, analysis of system design before it is being built and understanding of design change impacts*.

- RobMoSys enables systems *correct by construction*.

- RobMoSys supports management (design, assurance, traceability) of *(extra-functional) system properties* (e.g. resources, safety, QoS, accuracy, adequateness, etc.) in all development phases and at run-time:
  - deliver goods in time
  - trade-off energy consumption, speed, safety, etc.
Better Tools, Better Models, Better Systems: The Ideal Setting for Open Calls

- Address a relevant **use-case** (for system-level or run-time properties of composition) with coverage of **tooling** (not just paperwork/pipe dream), **models** (composable, architectural pattern, a design principle, an engineering model), and **systems** (prototypical scenario).
  - Use a **technology of another domain** and make it **accessible to robotics**. e.g. use timing analysis (e.g. from automotive), sanity checks at run-time, fail-operational modes, graceful degradation
  - We ask for contributions that illustrate a step-change in system-level composition for robotics and demonstrate them in real-world scenarios

- **Build tandems and benefit from complementary expertise**
  - e.g. software engineering + robotics
  - e.g. industry (SME, large industry, small-craft industry) + Research
  - e.g. robotics expert + application domain expert

- **Do not re-invent**, do not address topics in isolation BUT build on existing technologies (e.g. make them accessible or extend them), **build transformations between worlds**
Better Tools, Better Models, Better Systems: The Ideal Setting for Open Calls

- Address a relevant use-case (for system-level or run-time properties of composition) with coverage of tooling (not just paperwork/pipe dream), models (composable, architectural pattern, a design principle, an engineering model), and systems (prototypical scenario).

- Use a technology of another domain and make it accessible to robotics. E.g. use timing analysis (e.g. from automotive), sanity checks at run-time, fail-operational modes, graceful degradation.

- We ask for contributions that illustrate a step-change in system-level composition for robotics and demonstrate them in real-world scenarios.

- Build tandems and benefit from complementary expertise - e.g. software engineering + robotics - e.g. industry (SME, large industry, small-craft industry) + Research - e.g. robotics expert + application domain expert.

- Do not re-invent, do not address topics in isolation but build on existing technologies (e.g. make them accessible or extend them), build transformations between worlds.

Tentative Open Call Schedule

- Max. funding per experiment: 300,000 €
- Max. funding per partner in RobMoSys: 250,000 €
- Experiment runtime: 12 months
- Number of experiments: 12-14

Open Call 1
- Opens 07/2017
- Projects start 01/2018

Open Call 2
- Opens 04/2019
- Projects start 10/2019

Innovation Actions (IA)
- For non-profit organisations: 100% direct costs + 25% indirect costs.
- For profit making entities: 70% direct costs + 25% indirect costs.
Better Tools, Better Models, Better Systems: RobMoSys Pilot Examples
World Café

- **Table 1**: *RobMoSys Better Models, better Tools, better Systems: Building Blocks Supplier Perspective*
  - **Objective**: From the perspective of a building block supplier: how to make his building block composable and how to explicate and assure properties?

- **Table 2**: *RobMoSys Better Models, better Tools, better Systems: System Builder Perspective*
  - **Objective**: From the perspective of a system builder who composes systems out of building blocks: How to express and assure relevant system-level properties?

- **Table 3**: *RobMoSys Use-Cases, Scenarios and their Pain Points (Industry Perspective)*
  - **Objective**: What are Use-Cases/Examples that have an impact?

- **Table 4**: *Link between ROSIN and RobMoSys*
  - **Objective**: Q&A towards both projects and their relationship

General objective:
Better models, better tools, better systems.
- **What** are the topics?
- **How far** to go in these topics?