H2020—ICT—732410

RobMoSys

COMPOSABLE MODELS AND SOFTWARE FOR ROBOTICS SYSTEMS

DELIVERABLE 5.9:
FINAL REPORT ON OPEN CALL II EXPERIMENTS

Luz Martínez (TUM)

Project acronym: RobMoSys

Project full title: Composable Models and Software for Robotics Systems

Work Package: WP 5, Open Calls

Document number: 5.9

Document title: Final Report on Open Call II Experiments

Version: 1.0

Due date: 31/12/2020

Delivery date: 31/12/2020

Nature: Report (R)

Dissemination level: Public (PU)

Editor: Luz Martínez (TUM), Dayana Ramírez (TUM)

Author(s): Luz Martínez (TUM)

Reviewer: Marco Jahn (EFE)

[Please use the table below as an internal changelog. Delete before submitting the deliverable.]

Date	Name	Changes/Comments

Executive Summary 5

1 Introduction 6
1.1 Scope 7
1.2 Document Structure 7
2 Overall Activities and Achievements 8
2.1 ITP Management 8
2.2 Meetings and workshops 8
2.3 ITPs Introduction 8
2.4 ITP Coaching 10
2.5 Overall Progress Assessment 11
2.5.1 Overall Progress of Instrument # 1 11
2.5.2 Overall Progress of Instrument # 2 11
2.5.3 Overall Progress of Instrument # 3 13
3 Individual ITP Results 14
3.1 Instrument # 1: Individual ITP Results 14
3.1.1 Advanced Robot Simulations for RobMoSys (AROSYS) 14
3.1.2 RobMoSys adoption in the lighting system manufacturing environment (RoMan) 15
3.1.3 HRI Components for Assistive Robot (HRICAR) 16
3.1.4 Applying a model-based software platform to standardize the Rehab Robot (AMBSPSRR) 17
3.1.5 Mixed Reality Implementation within RobMoSys (MR4RobMoSys) 19
3.1.6 Mobile Inspection Robot Autonomy for Nuclear Decommissioning Authority (MIRANDA) 20
3.1.7 Underwater Robotic Simulator for RobMoSys (UWROSYS) 22
3.1.8 An Experiment in Applied Modular FOod Robots Applications (EXAMFORA) 24
3.1.9 Stereovision and radar for affordable and safe navigation (STERAS) 26
3.2 Instrument # 2: Individual ITP Results 27
3.2.1 Metacontrol for ROS2 systems (MROS) 27
3.2.2 Skill composition with verified system properties (SCOPE) 28
3.2.3 Formal Safety Analysis in Modular Robotic Applications (ForSAMARA) 30
3.2.4 Component Composition from Real-time Function Blocks (COCORF) 32
3.2.5 Verifiable Composition of Dynamics and Control Algorithms for Robot Motion (VeriComp) 34
3.2.6 Energy-Guided Control Stacks and Robot-Software Architectures using Model-Driven Design (RobMoSys-EGCS) 36
3.2.7 Safety Component Composition for Robots (SafeCC4Robot) 38
3.2.8 Composable Models for Compliant Interaction Control (CMCI) 39
3.2.9 QoS Metrics-In-the-loop for better Robot Navigation (MIRoN) 40
3.3 Instrument # 3: Individual ITP Results 42
3.3.1 Guidelines for Improving SmartMDSD with DDS and QoS attributes for communications (SmartDDS) 42
3.3.2 Guidelines for Improving Papyrus for Robotics with PDDL Planning: Planning4Papyrus (Planning4Papyrus) 43
3.3.3 Human-Robot Coproduction (HRC) 44

3.3.4 OPC UA for RobMosys (OPC UA for RobMosys) 45
4 Conclusions 47

Executive Summary

This document presents the final review of twenty-two Integrated Technical Projects (ITPs) selected in the second RobMoSys Open Call. This deliverable reports the progress of work package WP5 “Open Call” of the RobMoSys Project, related to task T5.9 Final Report on Open Call II Experiments.

The main topics of this report are:

- Background on Integrated Technical Project (ITP) management and expected objectives.
- Global assessment in terms of performed activities and achievements.
- Individual ITP partial reports and assessment.

This report focuses on presenting the contributions and current status of the ITPs chosen in the second RobMoSys Open Call. The second RobMoSys Open Call had three cut-off dates. On the first cut-off date, proposals for instruments 1, 2 and 3 were assessed.

Instrument # 1: Fast Adoption has as a main objective to develop RobMoSys-conformant pilots (industrial case studies) based on existing assets (software and tools from the RobMoSys ecosystem), or provide software components conformant to the RobMoSys pilots.

Instrument # 2: Ecosystem Challenges aims at strengthening RobMoSys ecosystem by developing a baseline that includes models, tools, components, architectural patterns. For instrument 2, the plan included choosing proposals on seven different technical topics: (1) ROS 2 and Model-Driven Software Development, (2) Functional composition inside components, (3) System level composition / safety, (4) System level predictability of properties, Navigation, (5) System level predictability of properties, Manipulation, (6) OPC UA Robotics, and (7) Open topic.

Instrument # 3: Innovation Expert Intake targets proposals from legal entities to offer their expertise as a service. RobMoSys wants to take them on board in order to push innovation and strengthen the RobMoSys community. Expert services offered focus on the following fundamental topics (1) Adoption Measures, (2) Digital Infrastructure, (3) Market Uptake, (4) Community Creation, (5) Academy Growing.

As a result of the first cut-off date, one proposal of instrument # 1, nine proposals of instrument # 2 and one proposal of instrument # 3 were selected. During the second cut-off date six and three proposals were selected for instruments # 1 and # 3, respectively. The chosen proposals became RobMoSys ITPs. During the third and exclusive cut-off date two proposals were selected for instrument # 3. More information about the selection in the second call is available in the Deliverable 5.5 Evaluation and Selection Report for Open Call II.

ITPs of instruments # 1 and # 3 have a runtime of six months each, while the ITPs of instrument # 2 are planned to achieve their goals during twelve months.

Conclusions of this deliverable are:

- Most of the ITP teams achieved their key objectives, milestones, Key Performance Indicators (KPIs) and deliverables scheduled.
- The ITPs of the second call are building up the RobMoSys community on top of the developments achieved by the ITPs selected during the first RobMoSys open call. The process of aligning their proposed solutions with the RobMoSys paradigm and toolchains has been challenging. However, once the tools are learnt, ITPs are able to work with them in a smooth way. For this reason, some projects have reported few delays on the submission of their deliverables.
- Personal meetings have been replaced by web-meetings due to the world-wide COVID-19 virus spreading. By the time of this report, this situation has not affected ITPs’ progress significantly.
- Individualized coaching by RobMoSys core partners has proven once more beneficial for ensuring the integration of ITPs’ proposal into the RobMoSys ecosystem. Coaching is highly appreciated by both coaches and ITPs. The RobMoSys community is also getting stronger. Several ITPs reported that they are collaborating with other second call ITPs to leverage their results.

1 Introduction

RobMoSys's main goal is to create and consolidate an EU Digital Industrial Platform for Robotics to establish a common methodology for software development, improve tools and foster interoperability by model interchange and composability. The RobMoSys approach aims at solving critical issues observed in the industry and will draw a migration path for a stepwise adoption of existing systems for interested early adopters. It will consist of a specialized set of players with both vertical and horizontal interaction levels, providing both widely applicable software products and software-related services. This ecosystem will be able to rapidly address new functions and domains at a fraction of today's development costs.

The RobMoSys Open Calls are an essential means to achieve this goal. The First RobMoSys Open Call, which was open from July 2017 to October 2017, has funded six Integrated Technical Projects (ITPs) with a focus on strengthening the RobMoSys platform with better metamodels, tools, and models. They cover core technical aspects of the RobMoSys ecosystem, including behavior modeling, Quality of Service (QoS) management, functional safety, communication and control, and robotics performance benchmarking.

The second RobMoSys Open Call was opened in February 2019 and was initially announced with two cut-off dates. The first cut-off call closed on 07.05.2019 for Instruments #1, #2 and #3. The second cut-off date was announced for 31.10.2019 and only applied to Instruments #1 and #3. It was then decided to open an exclusive cut-off date to effectively use RobMoSys available budget. The exclusive call was open exclusively for those proposals that were submitted to one of the first two cut-offs, but were rejected. This decision was made due to the shorter runtime of the funding period. Namely, ITPs of Instrument # 2 have a runtime of twelve months while ITPs for Instruments # 1 and # 3 have a runtime of six months. In total, the second RobMoSys Open Call had three cut-off dates. On the first cut-off date, proposals for instruments # 1, # 2 and # 3 were assessed. The second and exclusive cut-off dates called proposals for instruments # 1 and # 3.

At the first cut-off date, 21 proposals have been submitted through the open call portal and 11 proposals were selected. At the second cut-off date, 15 proposals were received and 9 of them became RobMoSys ITPs. At the exclusive cut-off date, 4 proposals were received and 2 of them were selected to become RobMoSys ITPs.

This deliverable reports the progress of work package WP5 "Open Call" of the RobMoSys Project, related to task T5.5: "Monitoring and review". The description of the second open call is provided in the deliverable 5.2: Open Call II Preparation Documents, whereas the selection process is covered in deliverable 5.5: Evaluation and Selection Report for Open Call II.

Cut-offs of the Second RobMoSys Open Call

Cut-off	Instruments	Open call dates	Results	Starting date
First	1, 2 and 3	01.02.2019 - 07.05.2019	-Submitted Proposals: 26 -Selected Proposals: eleven (one ITP for Instrument #1, nine ITPs for Instrument # 2, one ITP for Instrument # 3)	Exact dates differ between ITPs. All ITPs started during October and November 2019.
Second	1 and 3	01.08.2019 - 13.11.2019	-Submitted Proposals: 15 -Selected Proposals: eleven (six ITPs for Instrument #1, three ITPs for Instrument # 3)	Exact dates differ between ITPs. All ITPs started during April and May 2020.
Third	1 and 3	13.02.2020 - 13.03.2020	-Submitted Proposals: four -Selected Proposals: two (both for	01.05.2020

			Instrument # 1)	

1.1 Scope

This report is focused on presenting the final results of the ITPs selected during the second call on the three cut-off dates by portraying objectives, results and coaches' comments on ITPs contributions to the RobMoSys ecosystem.

1.2 Document Structure

The remaining of this document is organized as follows:

- Section 2 provides background on ITP management and expected objectives, as well as a preliminary global progress assessment in terms of performed activities and achievements.
- In Section 3, individual ITP technical reports and coaches' opinions about ITPs' general progress are presented.
- The conclusions are provided in Section 4.

2 Overall Activities and Achievements

2.1 ITP Management

The consortium ambition is to integrate not just the ITPs contributions into the body of knowledge of RobMoSys, but also the ITP teams themselves into the RobMoSys community. To meet this goal, an important condition is to achieve a close and meaningful interaction between these ITP teams and the RobMoSys consortium, beyond the pure monitoring of their progress.

In order to guarantee and maintain the strong integration of ITPs with RobMoSys activities, three mechanisms have been effectively set:

(1) fluid communication,
(2) active coaching,
(3) agile reporting.

Fluid communication with ITPs have been supported by an infrastructure composed of a Discourse forum, and a Tuleap repository and wiki. Such a communication configuration is used as a first proof of concept for the fully open communication infrastructure that will be released for the upcoming RobMoSys community.

An active coaching strategy has been developed from work package WP5 according to which individual members of the consortium have been assigned to each ITP. These so-called "coaches" were actively involved in the technical development of the ITPs taking the monitoring process a step beyond pure tracking of the project progress.

In order to achieve agile reporting, a digital monitoring platform was set up by TUM. This platform allows us to store project deliverables, bimonthly reports and keep track of milestones as well as offering a space for discussion about the project progress between the ITP members and the assigned coach. This platform was designed to ease the coaches' work in keeping track of their project progress and reporting.

2.2 Meetings and workshops

The Kick-off Workshop for the ITPs selected in the second RobMoSys Open Call - first cut-off date took place at the Technical University of Munich, Germany on 23. and 24.10.2019 where members of the RobMoSys Core Consortium shared all details on the RobMoSys approach and explained the monitoring process. During this two-day meeting, the ITPs had the opportunity to meet their coaches and exchange experiences with other members of their fellow ITPs. This approach has already proven to be fruitful since collaborations between projects have emerged in the past. The meeting continued in small groups where coaches and ITPs discussed technical and non-technical issues in the implementation of their proposals and details on the connection between their contributions and the existing RobMoSys meta-models. The event was very productive towards aligning views within and across different projects.

2.3 ITPs Introduction

The second open call had three cut-off dates, were 22 ITPs belonging to instruments # 1, # 2 and # 3 have been selected. Below these ITPs are presented.

2.3.1 Instrument #1

Acronym	Cut-Off Date	Name
AROSYS	1st	Advanced Robot Simulations for RobMoSys
RoMan	2nd	RobMoSys adoption in the lighting system manufacturing environment
HRICAR	2nd	HRI Components for Assistive Robot
AMBSPSRR	2nd	Applying a model-based software platform to standardize the Rehab Robot
MR4RobMoSys	2nd	Mixed Reality Implementation within RobMoSys
MIRANDA	2nd	Mobile Inspection Robot Autonomy for Nuclear Decommissioning Authority
UWROSYS	2nd	Underwater Robotic Simulator for RobMoSys
EXAMFORA	3rd	An Experiment in Applied Modular FOod Robots Applications
STERAS	3rd	Stereovision and radar for affordable and safe navigation

2.3.2 Instrument #2

Acronym	Cut-Off Date	Name
MROS	1st	Metacontrol for ROS2 systems
SCOPE	1st	Skill composition with verified system properties
ForSAMARA	1st	Formal Safety Analysis in Modular Robotic Applications
COCORF	1st	Component Composition from Real-time Function Blocks
VeriComp	1st	Verifiable Composition of Dynamics and Control Algorithms for Robot Motion
RobMoSys-EGCS	1st	Energy-Guided Control Stacks and Robot-Software Architectures using Model-Driven Design
SafeCC4Robot	1st	Safety Component Composition for Robots (SafeCC4Robot)
CMCI	1st	Composable Models for Compliant Interaction Control (CMCI)
MIRoN	1st	QoS Metrics-In-the-loop for better Robot Navigation (MIRoN)

2.3.3 Instrument #3

Acronym	Cut-Off Date	Name
SmartDDS	1st	Guidelines for Improving SmartMDSD with DDS and QoS attributes for communications (SmartDDS)
Planning4Papyrus	2nd	Guidelines for improving Papyrus for Robotics with PDDL Planning: Planning4Papyrus
HRC	2nd	Human-Robot Coproduction
OPC-UA for RobMoSys	2nd	OPC UA for RobMoSys

2.4 ITP Coaching

As part of the efforts towards integrating ITPs' contributions into the body of knowledge of RobMoSys and ITPs' members into the RobMoSys community, ITPs are coached by an assigned partner of the RobMoSys consortium. The technical coaching was actively supported by Dr. Dennis Stampfer (THUlm) and each of the ITPs was being individually coached as presented below.

Instrument #1

- AROSYS - Dr. Dennis Stampfer (THUlm)
- UWROSYS - Dr. Luz Martínez (TUM)
- MIRANDA - Mr. Bare Luka Žagar (TUM)
- MR4RobMoSys - Dr. Dennis Stampfer (THUlm)
- AMBSPSRR - Dr. Ansgar Radermacher (CEA)
- HRICAR - Mr. Sergi Garcia (PAL)
- RoMan - Dr. Dennis Stampfer (THUlm)
- EXAMFORA - Dr. Herman Bruyninckx (KUL)
- STERAS - Dr. Herman Bruyninckx (KUL)

Instrument #2

- MROS - Prof. Christian Schlegel (THUlm)
- SCOPE - Dr. Matteo Morelli (CEA)
- ForSAMARA - Dr. Huascar Espinoza (CEA)
- COCORF - Dr. Enea Scioni (KUL) / Dr. Marco Frigerio (KUL).
- VeriComp - Dr. Herman Bruyninckx (KUL)
- RobMoSys-EGCS - Dr. Herman Bruyninckx (KUL)
- SafeCC4Robot - Dr. Ansgar Radermacher (CEA)
- CMCI - Dr. Enea Scioni (KUL)/ Dr. Herman Bruyninckx (KUL)
- MIRoN - Dr. Dennis Stampfer (THUlm)

Instrument #3

- SmartDDS - Dr. Alex Lotz (THUlm)
- HRC - Dr. Selma Kchir (CEA)
- OPC UA for RobMoSys - Prof. Christian Schlegel (THUlm)
- Planning4Papyrus - Dr. Matteo Morelli (CEA)

The duties of these coaches include:

- to actively be involved in the technical development of their assigned ITP
- aligning with RobMoSys background and to contribute in a consistent way

- to be an external partner for discussions with a strong background in the RobMoSys methodology
- to serve as the main link between the ITP and the RobMoSys consortium for questions, requests or to trigger potential collaborations or interactions between ITPs.
- to work closely with ITPs in online conference calls and ad-hoc face-to-face meetings.

2.5 Overall Progress Assessment

2.5.1 Overall Progress of Instrument # 1

One ITP was selected during the first cut-off date to contribute to develop RobMoSys-conformant pilots (industrial case studies) based on existing assets (software and tools from the RobMoSys ecosystem), or provide software components conformant to the RobMoSys pilots. Another six ITPs were selected during the second cut-off date. The number of proposals selected during the exclusive cut-off date are not available in the period of this report. Detailed information and progress achieved by each ITP are described in section 3 of this report. Additionally, the results and the coaches assigned for each ITP selected during the second and third cut-off dates will be reported in this deliverable.

The work plans of the seven ITPs chosen so far focus on the following technical objectives:

a. Understand and reproduce locally RobMoSys tools such as the Assistive Mobile Manipulation pilot.
b. Integrate RobMoSys components such as Flexible Navigation Stack domain models, RobMoSys SmartSoft framework into already existing robots like Summit-XL Steel robot (ROBOTNIK), Hans Robot Elfin 5 (Canonical Robots) and software like GMV Brain autonomy software, BEAST robotic platform or Unmanned Underwater Vehicle Simulator (UUVSimulator).
c. Test and validate the robot in industrial environments. And assess the performance, maintainability and scalability across a computer cluster of RobMoSys components integration into existing software components. University of Ulm's pilot Intralogistics Industry 4.0 Robot Fleet Pilot, the CEA's pilot human-robot collaboration assembly and eITUS will be used for this purpose.
d. Document projects' contribution inside and outside the source code so that users understand easily how to adapt ITPs' result to different use cases. As well as recording step-by-step tutorials about how to use new software developments in the RobMoSys ecosystem and adding information to different platforms like RobMoSys wiki page, RobMoSys discourse, RobMoSys forum and GitHub.

During this period, the only ITP consortium working in the framework of instrument # 1 has proactively boosted the fast adoption of the RobMoSys paradigm in industry and have achieved the following results:

a. AROSYS ran and developed demonstration systems by interfacing external controller mechanisms of Webots with the SmartMDSD Toolchain. All the components and systems developed as well as the documentation explaining how to use and how to create new components and systems are available on the ITP's public GitHub repository. One video showing an intermediate demonstration is available on YouTube. This video was promoted on the communication channels of Cyberbotics (Twitter, LinkedIn, YouTube, Discord, etc.).

2.5.2 Overall Progress of Instrument # 2

Nine ITPs are developing the RobMoSys ecosystem baseline that includes models, tools, components, architectural patterns on seven different technical topics: (1) ROS 2 and Model-Driven Software Development, (2) Functional composition inside components, (3) System level composition / safety, (4) System level predictability of properties, Navigation, (5) System level predictability of properties, Manipulation, (6) OPC UA Robotics, and (7) Open topic. Their technical objectives are:

a. Connect the ROS2 and RobMoSys frameworks, and further enable a metamodeling-sound usage of ontologies for robot software architecting.
b. Provide tools that analyse and derive properties of a task by composing the properties that describe its skills and the environment, and, at runtime, ensure the correct execution of a task by

monitoring it and propagating anomalies detected at the level of the skills.

c. Define and test a methodology (process) for guided formulation of machine-readable safety properties derived from current collaborative robotics standards.
d. Generate RobMoSys compliant digital function block data-sheets containing formalized constraints and QoS properties using the microblx real-time function block framework.
e. Augment functional composition inside components with verifiable properties and domain-specific extensions, linking the latter to the system-level RobMoSys composition structures.
f. Use the energetic information of the components as status information of those components, such that higher-level control layers can infer whether the system operates as expected.
g. Develop a methodology and tool support for the creation of robotics components ensuring safety at system level.
h. Provide composable domain-specific models, model transformations and a controller implementation together with tools to easily synthesize the required motion control components for advanced robots.
i. Systematically use models for dynamically reconfiguring the robot behavior, defined in terms of Behaviours Trees, according to the runtime prediction and estimation of Robot Quality-of-Service (QoS) metrics defined on non-functional properties.

The following overall results have been achieved:

a. MROS developed a draft ontology from the analysis of the pilot navigation scenarios (Bosch, URJC). This ontology aims at supporting metacontrol system modeling and run-time reasoning.
b. SCOPE described in detail the validation scenarios, in terms of behavior trees, model of the environment, skills, components, their interaction in terms of communication channels and the required properties. SCOPE defined the demonstration scenario that will be considered in the project, including the system properties and described the interaction patterns among skills and interaction patterns between skills and other components at the functional level in terms of RobMoSys metamodels.
c. ForSAMARA implemented system safety requirements as properties of the demonstrator D1 which models a simple collaborative assembly use case with two different robot setups, a Panda robot from Franka Emika and a UR5 robot from Universal Robots. The ITP modeled and designed D1 using Papyrus for Robotics toolchain to describe the behavior of the robot and the associated workflows as well as the significant characteristics of the robot system. Specific parts of the models defined in Papyrus were then transferred into abstracted artefacts used for model checking that yielded a verifiable model of D1 consisting of eight source files which are merged under given dependencies.
d. COCORF described, implemented, tested and documented the DSL composition extensions and the digital data-sheet for function blocks developed during this reporting period. Examples are documented in the microblx repository.
e. VeriComp selected verifiable properties and identified constraints that must be checked by the tooling to assure a correct composition. From the identified requirements and domain models, they derived four formal meta-models for verifiable motion control algorithms, namely Algorithms; Geometry, mechanics and kinematic chains; Control and Timing meta-models. These meta-models are RobMoSys conformant.
f. RobMoSys-EGCS is developing the first version of the EGCS control stack meta-models and defining industrial use cases with two end-user partners, TNO and VIRO.
g. SafeCC4Robots discussed and defined 16 high level requirements of the SafeCC4Robots platform from the user point of view and provided twelve metrics that quantify the extent to which SafeCC4Robot's methodology and tools support the creation of components for robotics that ensure safety at system level during and after the composition.
h. CMCI adapted the already existing language modules of CoSiMA to conform to RobMoSys, modeled two stacks of tasks for humanoids applications performing manipulations based on the domain analysis that includes a prioritization model and a Qp model.
i. MIRoN gathered information about the potential variation points and adaptation policies to be showcased on the two RobMoSys pilots related to navigation, i.e., the Intralogistics Industry 4.0 Robot Fleet Pilot (provided by THUlm), and the Assistive Mobile Manipulation Pilot (provided by

PAL).

2.5.3 Overall Progress of Instrument # 3

By the time of this reporting period, there are four ITPs working to push innovation and strengthen the RobMoSys community on the following fundamental topics (1) Adoption Measures, (2) Digital Infrastructure, (3) Market Uptake, (4) Community Creation, (5) Academy Growing. One ITP was selected during the first cut-off date and the remaining three ITPS during the second cut-off date. Additionally, the results of and the coach assigned for each ITP selected during the second and third cut-off dates will be reported in the Deliverable 5.9 Final Report on Open Call II Experiments. Detailed information and progress achieved by each ITP are described in Section 3 of this report.

In general, ITPs of instrument # 3 have as technical objectives:

a. Include part of the DDS core standards in SmartMDSD to be able to use DDS as an alternative middleware.
b. Include a PDDL-based planner in the Papyrus for Robotics tool. This planner will be applied in the Human-Robot Collaboration for Assembly pilot, which CEA is carrying out.
c. Benchmark in close collaboration with the RobMoSys consortium ITPs' results regarding how the RobMoSys' goal of a composable system can be integrated with a model of ensured usability by the human operator, and if possible, contribute to this.
j. Extend the OPC UA capabilities of the RobMoSys project.

As main results, instrument #3 is progressing in the following way:

a. The SmartSoft/DDS implementation is now fully tested and is planned to be available with the next release of SmartMDSD. RTI Connext DDS licenses must be obtained before using the extension SmartDDS. Ulm University of Applied Sciences (HSU) has provided a reference implementation of the patterns using the RTI Connext DDS product. Currently we are working on the integration of the DDS-compliant RobMoSys components with other software ecosystems using DDS. SOSS, which acts as a common gateway between closely related middleware, is now able to connect DDS publishers and subscribers that run in different domains and security contexts, just using a configuration file with the specific attributes of the gateway behavior.

3 Individual ITP Results

The results presented in this section were extracted from the deliverables provided by each ITP and the feedback by their respective coach.

3.1 Instrument # 1: Individual ITP Results

3.1.1 Advanced Robot Simulations for RobMoSys (AROSYS)

ITP Advanced Robot Simulations for RobMoSys (AROSYS) is conducted by Cyberbotics S.à r.l.

The coach assigned for this ITP was Dr. Dennis Stampfer (THUlm).

Project objectives

AROSYS aims at integrating the Webots robot simulator into the RobMoSys framework to provide additional capabilities to users such as reproducibility, predictability, support of Windows, Linux and macOS, high-fidelity rendering, sensor accuracy, and stable physics engine. Specifically, the ITP will:

O1. Get familiar with the RobMoSys architecture at large. More specifically, we will understand and reproduce locally the RobMoSys Assistive Mobile Manipulation pilot. During this work, conducted in close relationship with the pilot maintainers, Cyberbotics will perform an in-depth review of this pilot.

O2. Integrate the RobMoSys SmartSoft framework by replacing the inter-process communication (IPC) mechanisms, i.e., local pipes and shared memory segments that Webots uses for communicating with robot controllers, by the more standard RobMoSys SmartSoft framework. We will evaluate the impact of this change with respect to performance, maintainability and scalability across a computer cluster.

O3. Fully integrate the RobMoSys Assistive Mobile Manipulation pilot studied in T1 within the new version of Webots developed in T2, using the SmartSoft framework. This Webots-based pilot will be a proof-of-concept that could be modified and extended by users. Therefore, it will be very well documented (inside the source code), so that users understand easily how to adapt it to different use cases.

O4. Document our contribution in this project in the official Webots documentation. A tutorial explaining step-by-step how to use the Webots software within the RobMoSys architecture will be added to the RobMoSys wiki pages.

O5. Create and publish a promotional video to advertise the results of the project and to encourage the RobMoSys community and other robotics communities to integrate Webots + RobMoSys into their workflow.

Reported results

O1. The ITP coordinator reported that it took them quite a long time to get familiar with the SmartSoft MDSD Toolchain. However, once the team understood the way to use it and extend it, it became very easy to run all the demonstration systems and develop their own ones.

(Completed: 100%)

O2. A new version of Webots containing all the required improvements to Webots for interfacing with the SmartMDSD Toolchain (mainly improvements related to the external controller mechanism) was released on the 14th of January. It can be downloaded here:
https://github.com/cyberbotics/webots/releases/tag/R2020a-rev1

(Completed: 100%)

O3. All the components and systems developed for this project are available on this public GitHub repository: https://github.com/cyberbotics/AROSYS

(Completed: 100%)

O4. The documentation explaining how to use the components and systems created and how to create a new one is available directly on the wiki of the GitHub repository of the project: https://github.com/cyberbotics/AROSYS/wiki The list of components and systems available out of the box is visible in the root README file of the GitHub repository of the project: https://github.com/cyberbotics/AROSYS/blob/master/README.md

(Completed: 100%)

O5. The video showing intermediate demonstration is available on YouTube: https://youtu.be/7rNtWxuXHKs This video was promoted on the communication channels of Cyberbotics (Twitter, LinkedIn, YouTube, Discord, etc.).

(Completed: 100%)

Coach's perspective

After an initial learning phase - which was quite short - the ITP was very quick in implementing the use-case. The coach had very close interactions with the ITP and therefore was very aware of all activities, issues and developments. The project has met its objectives. The project and the chance for dissemination and uptake through the simulator is quite a benefit for RobMoSys.

3.1.2 RobMoSys adoption in the lighting system manufacturing environment (RoMan)

ITP RobMoSys adoption in the lighting system manufacturing environment (RoMan) is conducted by Robotnik Automation SLL.

The coach assigned to this ITP was Dr. Dennis Stampfer (THUlm).

Project objectives

Integrate the Flexible Navigation Stack domain models, a Tier 2 domain structure, in the Summit-XL Steel robot manufactured by ROBOTNIK. With the integrated Tier 2 domain structures, ROBOTNIK will assume the System Builders role and compose the industrial application with the provided Tier 3 mapping and localization services. This integration of RobMoSys components will increase the modularity and generalization of the service robots and validate the reusability of the previously created components in a completely new robot. The RobMoSys structure will ensure a proper integration of the robot in a lighting systems manufacturing company. Specific objectives for this project are:

O1. Adapt the existing robot software to integrate RobMoSys components.

O2. Develop the application and integration of the robot in the industrial company systems and network.

O3. Test and validate the robot in the real lighting systems manufacturing installations of the company. Once the autonomous navigation in the facilities is verified, there will be a short training to the company workers to teach them how to command the robot remotely from their own system.

Reported results

O1. RobMoSys components allowed RoMan to command navigation goals of the Summit-XL Steel robot through different Navigation Stack components.

The ITP designed a scheme to integrate the Flexible Navigation Stack with the Summit-XL Steel robot. The integration allowed RobMoSys components to access the laser data of the robot and send velocity commands.

Components were validated individually before the integration. Parameters and footprint were redefined to achieve a successful obstacle avoidance. The redefinition was successfully tested and visualized with the RobMoSys visualization tool.

(Completed: 100%)

O2. Using the developed and existing components of the RobMoSys environment it was possible to perform the necessary tasks for a commissioning in an industrial environment with the Summit-XL Steel robot. Including and integrating the needed components :

- Map an industrial installation
- Verify the wireless communication with the robot
- Navigation tests between the production area and the warehouse.

Wireless communication couldn't be optimized and be executed in a more user friendly interface. Robotnik had to make use of the open Software UAExpert.

(Completed: 85%)

O3. The robot was successfully implemented in the company. It was moving load between the two desired areas. It was remotely called through an available PAD or computer. Due to COVID restrictions only two workers of the company were trained and able to command the robot (one per shift).

(Completed: 80%)

Coach's perspective

A first running demonstrator was provided just a few days after the project start. The speed with which the project was carried out was very fast. The ITP has continuously updated the baseline scenario and integrated new components. The coach was regularly informed on such updates and provided with video material. In a final video, it has been demonstrated that the scenario was fully implemented and working in an industrial environment.

3.1.3 HRI Components for Assistive Robot (HRICAR)

ITP HRI Components for Assistive Robot (HRICAR) is conducted by Advanced Deep Learning Robust Basic Transactional Services SL (ADL RBTS SL).

The coach assigned to this ITP was Mr. Sergi Garcia (PAL).

Project objectives

ADL RBTS SL aims to develop a set of RobMoSys models and components for human-robot interaction, and adapt some components of their AI cloud platform FIONA to make it RobMoSys compliant. Once the developed components are available, they will integrate the developed components into the Assistive Mobile Manipulation Pilot. Specifically, ADL RBTS SL will focus on the use case of creating assistive applications for TIAGo.

O1. Adapt the models for human-robot interaction from FIONA to RobMoSys.

O2. Develop human-robot interaction components for the SmartSoft World by adapting the components face detection, face expression recognition and human body pose recognition, already running in FIONA. These human-robot interaction components will comply with and validate the models adapted in the first objective.

O3. Integrate the human-robot interaction components into the Assistive Mobile Manipulation Pilot and design examples.

Reported results

O1. The existing available models were more generic than anticipated. The initial objectives for ADL RBTS considered the extension of the existing models due to the expertise in modelling of the company. After analysing the available models, ADL RBTS considered that the existing one fulfilled all the requirements or developed the components.

(Completed: 100%)

O2. Three components were developed by ADL RBTS SL: face recognition, expression recognition and body pose-recognition. The face-recognition implements a state-of-the-art algorithm which achieves a 99.38% accuracy on the Labelled Faces in the Wild Benchmark. The expression recognition identifies 7 basic human emotions in real time. The model will generate seven probability values corresponding to seven expressions. The highest probability value to the corresponding expression will be the predicted expression for that face. For the body pose-recognition a set of 2D confidence maps for the body part locations are generated. There are 19 confidence maps: 18 of them are body parts and the 19th is the background. Body pose recognition code has a bigger database than face and emotions codes and also needs much more processing so due to the lack of GPU, the tests show a 95% accuracy and pose recognition (standing or sitting) has about 80% accuracy.

(Completed: 100%)

O3. The three components were integrated into the virtual environment of the TIAGo robot for the Assistive Mobile Manipulation Pilot. The components can be deployed on a simulation environment or in the real robot with or without graphical acceleration. The development environment includes the SmartSoft toolchain and the TIAGo utilities provided in an off-the-shelf virtual machine.

(Completed: 100%)

Coach's perspective

The existing SmartSoft models were useful because they were used as is with no modifications for component development. On the one hand, it would have been interesting to have contributions in the tier- 2 domain due to the partner's experience. On the other hand, it demonstrates the integrity of the models provided in the area of human interaction.

The HRICAR ITP contributes to improving the RobMoSys components already available in the area of human-machine interaction, which is essential for the capabilities of service robots, especially in healthcare settings. Most pilots do not focus on human-machine interaction, so this project provides examples in an unexplored area of RobMoSys. In relation to the developed components, the recognition metrics achieved are good compared to the state of the art. It would be interesting to expand the number of emotions to detect and to detect lying bodies. As a recommendation to improve human-machine interaction, it would be interesting to add an automatic speech recognition component and a text-to-speech component so that they could be used in health care applications.

The integration of the components was done in the TIAGo robot simulation as planned from the Docker container provided by PAL Robotics. This fact demonstrates the maturity and usability of the tools provided.

3.1.4 Applying a model-based software platform to standardize the Rehab Robot (AMBSPSRR)

ITP Applying a model-based software platform to standardize the Rehab Robot (AMBSPSRR) is conducted by Canonical Robots S.L.

The coach assigned to this ITP was Dr. Ansgar Radermacher (CEA).

Project objectives

AMBSPSRR has as a main objective to adapt some of the functionalities implemented in the "Rehab

Robot", related to human-robot interaction, to develop a RobMoSys-conformant pilot based on existing assets (software and tools from the RobMoSys ecosystem).

O1. Learn the theoretical concepts applied in the CEA_Pilot and eITUS.

O2. Select some components from "Rehab Robot" to be adapted to RobMoSys.

O3. Test and enrich RobMoSys safety functionalities with Hans Robot Elfin 5 collaborative robot performing a path movement task. AMBSPSRR will test and validate the solution and provide feedback for enhancements during the project lifetime and after the project. The demonstrator is elaborated based on the CEA_Pilot and eITUS.

Reported results

O1. AMBSPSRR have studied the RobMoSys modeling concepts using the available public documentation. We had also gained ground knowledge and understanding of the theoretical concepts applied in CEA_Pilot and eITUS and compared SmartMDSD and Papyrus for Robotics toolchains from the perspective of their role as Component Suppliers who shall conform to Tier 2 meta-models in the implementation of any component involved in the proposed "Rehab Robot" use case.

During this process we determined that SmartMDSD toolchain is more suitable for our needs. So we have decided to use this tool instead of Papyrus.

(Completed: 100%)

O2. After studying the different subsystems of the "Rehab Robot", the component for constructing 3D models using a depth sensor and a structure from motion approach. In order to select this component, we took into consideration the value it brings to the platform and how it integrates with other RobMoSys components. This was done with the invaluable help of our coach.

The component that AMBSPSRR has developed in the WP3 is the novel "Component3dReconstruction". This component has been introduced in order to perform real-time 3D surface reconstruction based on depth sensor data, implementing the popular computer-vision algorithm "Kinect Fusion".

This component is an essential part of the "Rehab Robot" because it is used to obtain the 3D model of the patient's body and has a wide range of potential applications outside of this project.

(Completed: 100%)

O3. We have enriched RobMoSys by developing the component "Component3dReconstruction" that is hosted and publicly shared on GitHub:

https://github.com/canonical-robots/RobMoSys-AMBSPSRR

In the second deliverable of the working package 3 we have tested the component performance and integration in the RobMoSys ecosystem.

A new service was introduced on our request in the composition Tier 2 vision-related domain that has also enriched the RobMoSys framework.

(Completed: 100%)

Coach's perspective

The project developed a new component for the RobMoSys tooling, the 3D reconstruction based on point cloud data. The AMBSPSRR results were provided in form of an implementation for the SmartMDSD toolchain.

The results are useful for follow-up projects and for direct usage in concrete use cases with a similar setting and demonstrate that the RobMoSys approach can be quickly adopted by companies, even if not experienced with a model-driven approach.

3.1.5 Mixed Reality Implementation within RobMoSys (MR4RobMoSys)

ITP Mixed Reality Implementation within RobMoSys (MR4RobMoSys) is conducted by Awesome Technologies Innovationslabor GmbH.

The coach assigned to this ITP was Dr. Dennis Stampfer (THUlm).

Project objectives

This ITP aims at using an existing Bridge between ROS industrial and the game engine Unity developed in a ROS_In FTP in order to implement an AR/VR visualization in the context of University of Ulm's pilot "Intralogistics Industry 4.0 Robot Fleet Pilot". They see four different application scenarios for AR/VR technology within the context of a smart logistics system: i) Installation ii) Interaction with workers at the factory floor iii) maintenance and troubleshooting and iv) supervisor. MR4RobMoSys want to develop their existing framework further with the help of the SmartMDSD approach and Papyrus for Robotics.

O1. Define priorities and create the backlog for the agile development of an AR/VR technology and learn about SmartMDSD and Papyrus for Robotics.

O2. Develop a "potentially shippable increment" of the final product by using a SCRUM-based agile method which works in sprints.

O3. Disseminate results using videos and written reports.

O1. We defined a backlog including (First Priority):

- Running SmartMDSD
- Sending UDP packages from SmartMDSD to Unity -
- Evaluation WebXR versus Unity
- Showing geometric primitives in Unity
- Minimal viable prototype based on recorded information
- Discussion: Converting driving commands in 3D objects

Second Priority

- Getting driving intentions of the robot (how it will move in the next seconds)
- Visualisation of next driving intentions of the robot.
- Visualisation of laser points
- Visualisation of paths from the path planning, incl. corridors in which robots move
- Virtual teaching of paths

(Completed: 100%)

O2. We identified the second priority points from above as an increment of the product. These could be realised using a SCRUM-based agile method.

(Completed: 100%)

O3. We showed all first priority points in a 4-minute Video. In it one can see a robot moving and the path of the robot overlaid using Augmented Reality (AR). A robot in a simulation was also utilised for testing purposes. We also wrote a technical wiki page to explain our project. There we discuss how we integrated the Unity-based AR-App with the RobMoSys Components and Systems.

(Completed: 100%)

Coach's perspective

The project has maintained excellent communication with the coach. There were short loop interactions and discussions on how to proceed at each intermediate step during development. After an initial learning-effort, the project has gained a lot of speed and implemented two very useful software components for the RobMoSys ecosystem: the visualization of movement intentions of a mobile robot and the visualization of laser scan readings. The project has demonstrated the developments both in augmented and virtual reality on their site. The whole software stack was set into operation at a third party site without hassle without the support of the developers/project. This shows that a sufficient maturity level has been achieved.

3.1.6 Mobile Inspection Robot Autonomy for Nuclear Decommissioning Authority (MIRANDA)

ITP Mobile Inspection Robot Autonomy for Nuclear Decommissioning Authority (MIRANDA) is conducted by GMV Innovating Solutions

The coach assigned to this ITP was Mr. Bare Luka Žagar (TUM).

Project objectives

The objective of the MIRANDA project is to use RobMoSys with existing GMV hardware and software to produce a system capable of running a typical field trial in the context of a nuclear site. This demonstration requested by the Nuclear Decommissioning Authority (NDA) will allow GMV to bring together WatchChainR and ARISE developments, access a new user for the technology and create RobMoSys compliance. The GMV Brain autonomy software, and the BEAST robotic platform, currently undergoing development and field trials, will be integrated using the RobMoSys interface design tools. Each will be represented by separate RobMoSys components. In this way, the GMV software, both autonomy and platform-specific components, can be used to evaluate the RobMoSys approach and structures. GMV will use existing developments with the aim of creating a long-term product with RobMoSys compliance. Planned use cases are external pipeline inspection and inspection in mining.

O1. Analyse existing RobMoSys Components and Intralogistics Industry 4.0 Robot Fleet Pilot. Analyse the RobMoSys ecosystem for appropriate components that will complement GMV Brain in the nuclear inspection use-case, and inspect the existing pilot being used as a base-line.

O2. Use SmartMDSD tool to model proposed system components in a RobMoSys compliant way.

O3. Design the interface between BEAST, GMV Brain and RobMoSys and compose the system from these existing components using SmartMDSD.

O4. Develop the GMV Brain RobMoSys wrapper for GMV Brain components.

O5. Implement the BEAST RobMoSys wrapper for the BEAST platform. It should act as a bridge for all communications from the low level robot components, isolating only those necessary and providing the RobMoSys components with the required data.

O6. Integrate GMV Brain, BEAST and any additional RobMoSys component into a single system.

O7. Perform integration testing in simulation and in the lab.

O8. Run the field trial at the test site. It is proposed to perform a preliminary test to be allocated to the first week and the final test allocated to the final week. This gives a week to react to any issues, such as poor weather or mechanical problems with the platform.

Reported results

O1. The analysis of the RobMoSys ecosystem and the Robot Fleet Pilot components was made in the context of MIRANDA, that being an outdoor infrastructure inspection scenario. As the BEAST software was already complete, an improvement of existing functionality was a requirement for any components to be integrated into the system. The analysis was performed by targeting components that support the integration of GMV Brain, such as battery state analysis and additional sensor support. It was found that the majority of components target simulators other than the ones used by GMV, specific sensors, or were developed for indoor applications. The SmartLaserLMS200Server component was identified as a potential candidate for integration, but was ultimately not included due to decisions made regarding the platform on other activities. Next steps for this objective would be to use the Flexible Navigation Stack and P4R to replace and compare the current BEAST camera-based mapping and the planning components with those distributed as part of SmartMDSD. In this way, we would determine if planners designed for use in corridors can be effectively used in a hybrid artificial/natural environment.

(Completed: 100%)

O2. After the initial project coach meeting, it was decided that it would be possible to change the target middleware from ROS to ROS2.0 and so use the Papyrus4Robotics tool. Doing so enabled GMV to evaluate the current status of ROS2.0 and evaluate P4R. This tool was then used to model the BEAST and GMV Brain wrappers. Additional work was done to model the BEAST system using the tool. While BEAST currently operates using ROS, not ROS2.0, is was a useful exercise in using the tool to document the required interfaces and allowed for a more complex system to be modelled.

(Completed: 100%)

O3. The interface between BEAST and GMV Brain was designed using the Papyrus4Robotics tool, after creating the additional ROS2.0 message definitions required by GMV Brain. The ros1_bridge for integrating the ROS and ROS2.0 components was selected to ease the data transfer. Eventually we want to transition to ROS 2.0 interfaces only.

(Completed: 100%)

O4. The GMV Brain Wrapper component utilised the code generation functionality of the P4R tool to create a basic node with the appropriate ROS2.0 input and output topics and actions. This node was then modified to correctly handle incoming and outgoing data.

(Completed: 100%)

O5. The BEAST Wrapper was implemented in ROS to handle incoming and outgoing data. It was modelled in P4R to match the data requirements of GMV Brain, utilising building ROS versions of the ROS2 message types developed for the GMV Brain output goals and status topics.

(Completed: 100%)

O6. Integration of the BEAST, GMV Brain and RobMoSys wrappers was performed on a desktop running an Ubuntu 18.04 environment. ROS 2.0 Eloquent was used as part of the requirements for Papyrus4Robotics and ROS Kinetic for the BEAST system.

(Completed: 100%)

O7. Integration testing was performed using the Gazebo simulator for BEAST that was developed in previous activities, and confirmed the resulting behaviour as correct compared to previous field demonstrations.

(Completed: 100%)

O8. Due to COVID restrictions, the field trial had to be cancelled changed to a simulation only test. As a replacement, further iterative testing was performed in the simulated environment. While not having field trial is unfortunate in simulation much more time was spent on iterative development than would have been possible in the field which is beneficial for system integration activity. Due to issues with accurate resource (robot power levels) estimation by the on board planner, plan failures had to be dropped from the list of tests to be performed. However, multiple-goal plans were successfully performed and the behaviour of the system in response to dynamic goal injection was confirmed.

(Completed: 50%; Most of the original tests were performed, physical field trial was not achievable)

Coach's perspective

MIRANDA's progress has been constant and smooth during the entire project duration. The communication with the coaches was smooth and regular, so the progress could be easily monitored. The ITP team participated in the ITP kick-off meeting to present their goals and in the ITP virtual workshop to present results.

GMV Brain has been successfully integrated with the BEAST software and tested within the context of a simulated environment, and developed the software interfaces using the Papyrus4Robotics tool, thus extending the scope of the tool to ground inspection applications. As a result of this ITP a demonstration video was made. Finally, all the objectives and deliverables of the ITP have been fully accomplished.

Due to COVID restrictions, the field trial had to be cancelled and changed to a simulation only test.

3.1.7 Underwater Robotic Simulator for RobMoSys (UWROSYS)

ITP Underwater Robotic Simulator for RobMoSys (UWROSYS) is conducted by Skarv Technologies AS. The coach assigned for this ITP was Dr. Luz Martínez (TUM).

Project objectives

UWROSYS will bring a high fidelity underwater robotic simulator with advanced sensors into the framework of RobMoSys, opening the underwater domain to RobMoSys users, by building on the Gazebo/TIAGo/SmartSoft Scenario and the Intralogistics Industry 4.0 Robot Fleet Pilot. UWROSYS will build on the existing open source Unmanned Underwater Vehicle Simulator (UUVSimulator) that is an open-source simulator based on Gazebo, which allows simulation of several types of UUVs. UWROSYS will be made configurable and will be made compatible with several of the existing SmartMDSD toolchain blocks.

O1. Familiarize with the SmartMDSD-framework. Test and setup existing SmartMDSD modules. Assess feasibility of reusing modules directly or if adaptation is necessary. Evaluate possible future extensions.

O2. Integrate UUVSimulator into SmartMDSD by adopting SmartGazeboBaseServer. Implement remote control of ROV using SmartMDSD joystick module.

O3. Modify UUVSimulator SmartMDSD integration to include manipulator arm interface, and remote control of ROV manipulator arm using SmartMDSD joystick module.

O4. Integrate camera view in UUVSimulator SmartMDSD integration for future use with SmartMDSD object recognition modules. Evaluate possibility of using virtual reality.

O5. Integrate ROV sonar for use with existing SmartMDSD-modules such as obstacle avoidance modules (i.e. SmartCdlServer).

O6. Report and generate a demonstrator video. Publish a Docker or virtual machine demo.

Reported results

O1. The ITP investigated the different modules of RobMoSys, and the existing implementation of Gazebo in RobMoSys, both through code reading and testing of the existing system demonstrations. The existing Gazebo-implementation did not work for UWROSYS, due to version issues (different versions used in RobMoSys and ROS). The development team ended up selecting the 'Mixed-Port for ROS'-component for implementing the UUVSimulator in RobMoSys.

UWROSYS presented their goals and results at the kick-off workshop and the ITP workshop respectively.

(Completed: 100%)

O2. The UUVSimulator was implemented through the use of the 'Mixed-Port for ROS'-component (instead of the planned use of SmartGazeboBaseServer), where the input to the component is the commanded velocity of the robot, and the output is the state of the robot.

Remote control was implemented using the example code for the *ComponentRosJoystick*. Alterations were also made to this example code and the Domain-models to allow control of the robot in the 3D environment. This component was used in conjunction with the *SmartJoystickNavigation* (which were also modified to allow control in a 3D environment).

(Completed: 100%)

O3. The ROV manipulator arm was implemented in its own component, *ComponentUUVManipulator*. This component was also based on the 'Mixed-Port for ROS'-component, since this component proved to be a good way of implementing ROS-functionality in SmartMDSD. Input to the *ComponentUUVManipulator* is the commanded manipulator arm velocity.

Remote control was implemented using the same *ComponentRosJoystick* used to command the UUVSimulator. It was decided that this was best to allow an operator to easily control both the vehicle and the manipulator arm. Control between the vehicle and the manipulator arm is switched by pressing a configurable button. In manipulator arm mode, the UUV will perform station-keeping (i.e. commanded velocity is zero).

(Completed: 100%)

O4. The camera-feed from the ROV camera was made available to SmartMDSD using the ComponentRosImageServer. This image-server was also implemented using the 'Mixed-Port for ROS'-component, and the output of this component is the operator video feed in the form of a CommVideoImage-object. An image-client, ComponentRosImageClient, was also implemented to view the video-stream through SmartMDSD.

Implementing virtual reality was considered outside the scope of this ITP, however, it is believed to significantly improve an operators immersive experience of the ROV operation. The simulator is quite computationally heavy in its current state, so computational complexity and especially lag (which can make the experience quite unpleasant for an operator) must be taken into consideration.

(Completed: 100%)

O5. The UUV sonar was implemented using the 'Mixed-Port for ROS'-component in a component named ComponentUUVSonar. The input to this component was the state of the UUV (received from the ComponentUUVSimulator), and the output is the sonar measurements (in a LaserService-object).

Implementation of the sonar measurements provide the possibility to utilize existing components such as obstacle avoidance and path planning modules (currently not been tested).

(Completed: 100%)

O6. Reporting and generation of demonstration videos were done accordingly. A significant effort was made to generate a Docker image that would automatically deploy and run the demo, however, due to some incompatibilities between Docker and SmartMDSD, this effort was abandoned. However, a virtual machine for running the UWROSYS-demo was generated.

(Completed: 100%)

Coach's perspective

The progress of UWROSYS has always been smooth and on time during the entire duration of the project, as well as the relation with the coaches. In addition, the ITP team participated in the ITP kick-off meeting to present their goals and in the ITP workshop in M05 to present partial results.

UWROSYS successfully achieved its goal of integrating a new simulation platform within the SmartMDSD toolchain, extending the scope of the toolchain to underwater applications. The UWROSYS simulator was integrated with new sensors, such as, joystick, manipulator arm, camera and sonar.

As a result of this ITP a virtual machine for running the UWROSYS-demo was generated and demonstration videos were made public (https://skarvtech.sharepoint.com/:v:/s/Skarv/EQ6HDRESlpdOqfbc317UCRsBaoiBlPp7tj8UapQmMAAM9g?e=EXR17S). Also, all the objectives and deliverables of the ITP have been fully achieved.

3.1.8 An Experiment in Applied Modular FOod Robots Applications (EXAMFORA)

ITP An Experiment in Applied Modular FOod Robots Applications (EXAMFORA) is conducted by R U Robots Limited.

The coach assigned for this ITP was Dr. Herman Bruyninckx (KUL).

Project objectives

The EXAMFORA project is an investigation of the use of the RobMoSys method within a hard real-time system (The GRAIL robot). The application is for a vision-based system that increases the overall efficiency of an assembly system involving human-robot interaction by choosing paths that are least likely to result in collisions. If successful, the RobMoSys approach will be applied to other parts of the GRAIL system during an upcoming port to a new industrial robot platform. Although no time savings are expected in this first port, considerable time savings would be expected in subsequent ports. 6. The application in question, the GRAIL Robot System, provides the capability to assemble sandwiches, pizzas, and other food products on an assembly line and can work alongside human co-workers.

O1: Learn about RobMoSys by conducting a familiarization activity.

O2: Implement RobMoSys components and wrappers in the GRAIL Robot system

O3: Evaluate the GRAIL / RobMoSys modules for Human Sensor Interface and Human Behaviour Prediction

O4: Create a video demonstrating the use of the new RobMoSys compatible components in a mock-up of a sandwich assembly application

Reported results

O1. The familiarisation study was carried out throughout the project as it was found that the learning curve was quite steep and implementing the different modules provided challenges that required further investigations of the RobMoSys approach and, particularly, the tools. In the end it was possible to undergo two iterations of module development with both the skeleton tracker (each using a different sensor input) and the human behaviour predictor (a simple one and a more "realistic" one that considered the kinematic constraints of the human body). Over these two iterations we observed definite speed ups as a result of familiarisation, but we still found the process somewhat more difficult than our standard development methods, probably more down to the lack of maturity of the tool, rather than anything else.

Overall we found the RobMoSys approach a useful one at the early design stages of a project and we will probably incorporate many of the concepts in our design methodology (which is already based on a modular approach with our own control system "library") At this stage we will not, however, be adopting any of the RobMoSys tools as these do not bring us significant benefit over our existing development approach and tools.

(Completed: 100%)

O2. Overall 5 new modules were developed within the project. These were:

- Two human skeleton tracker modules. One version was developed to work with an existing Kinect interface module and one with an existing Intel RealSense module. The change between the two was relatively straightforward, although there were inconsistencies between the two sensor interface modules which made the process more involved that was necessary (e.g. the outputs were different data formats and the units of measurement were different, on top of the expected resolution differences).
- Two human behaviour predictor modules. The first version used a simple model that simply looked at the velocity of the nearest nodes of the human skeleton and extrapolated the potential position at each time step. The second version used a kinematic model of the human skeleton to put physical limits on the extent of the possible travel. Although the two modules used very different internal coding, the interfaces were identical and thus the full advantage of the RobMoSys modular approach was gained.
- A plan chooser module. This module is very specific to the operation of the GRAIL robot system and chooses an optimum path for the robot based on the likely incursion of the human into the robot workspace. Because of the specificity of the application, this is the only module that will not be released into the public domain, however it was necessary for the completion of the full system developed under the RobMoSys system.

In addition, a wrapper was developed for the rest of the GRAIL robot system that handled all the communications between the developed modules and the GRAIL robot control system.

(Completed: 100%)

O3. When the project was proposed it was envisaged that interface modules would have to be developed for both the Kinect sensor and the Intel RealSense sensor. However, by the time the contract was awarded it became apparent that interface modules existed for both of these sensors. However, both of these modules output basic data, rather than identifying the presence and position of humans in the scene, which was needed for the GRAIL robot. It was therefore agreed that instead of developing sensor interface modules that we would use the existing modules but develop a skeleton tracker that would take the sensor interface module output, find humans in the scene, fit a skeleton and output the position and velocities of the skeleton nodes. This was done and was interfaced to each of the sensor interface modules, but as noted above, some changes to the interfaces were required to cope with the lack of standardisation between these modules. Thus with this "change of sensor" scenario the full benefits of the RobMoSys system were not realised.

With the human behaviour tracked, as also noted above, the "module upgrade" scenario was tested and this yielded much better results in terms of the ability of being able to update the internals of a single module without changing any other aspect of the system or the module interfaces.

(Completed: 100%)

O4. When the project was originally proposed it was intended to produce a video of the hardware operating with software developed using the RobMoSys approach and the RobMoSys tools. However, due to coronavirus restrictions, it was not possible to access the lab to undertake a full hardware implementation of the system. It was therefore decided to produce a video of a simulation of the GRAIL system with RobMoSys produced modules in the loop. This simulation is driven by the real GRAIL robot hardware and takes input from real sensors. In this sense it is only the movement of the robot that is simulated. This video has been produced and is available.

(Completed: 100%)

Coach's perspective

The company put three very senior, experienced robot system developers in the project, and that was a dream to interact with. These people were already applying many of RobMoSys' best practices and patterns, without realising it explicitly. The outcome is that the company has been able to harmonize its design approach, and is going to refactor things a bit more different in the future, with RobMoSys' longer term composability guidelines in mind. Interactions were very professional and efficient.

3.1.9 Stereovision and radar for affordable and safe navigation (STERAS)

ITP Stereovision and radar for affordable and safe navigation (STERAS) is conducted by Avular B.V.

The coach assigned for this ITP was Dr. Herman Bruyninckx (KUL).

Project objectives

Avular is developing a low-cost navigation system that is capable of simultaneous localization and mapping (SLAM). This allows mobile robots and drones to navigate autonomously and safely through a dynamic territory in the presence of dynamic objects such as other robots, humans, or cars. The navigation system is based on the fusion of a radar- and stereo vision-sensor through artificial intelligence (AI).

Avular aims to prove software safety by following the RobMoSys methodology to develop the software. The lower price of this navigation system opens up new markets for the exploitation of mobile robots in a range of verticals. Avular aims to use the RobMoSys methodology to make an important step towards proving software safety. For this, the ITP will establish a RobMoSys conformant formal framework of the fusion software and use statistics derived from in-field measurements to reliably predict the accuracy of the resultant depth map. This output along with the software architecture can be used to give a first estimate of the safety integrity level (SIL) of the software.

In the RobMoSys ecosystem, Avular is a user that acts as a component supplier to create the software and hardware of the navigation system. Any system builder in any discipline can use this software and the corresponding safety engineer can model the limits and constraints for the use of this navigation system for his/her application.

O1: Model the sensor setup

O2: Create a performance report of the navigation solution

O3: Demonstrate the RobMoSys conformant software

O4: Write a safety report

Reported results

O1. A lot of work has been done on this objective. We extensively studied different machine learning implementations for sensor fusion. This turned out more difficult than expected, as there are not many examples of machine learning implementations that work in real time on embedded software available in literature. Eventually we identified a machine learning network that has shown to be an efficient network for real time depth map estimation using stereo images as input. We adapted this network to be able to also process radar data and trained this network with a custom dataset and a publicly available dataset. Based on this network, we were able to create several formal UML models of our solution for application in a mobile robot, in relation to a health monitor and in a safety loop. The models are a confidential deliverable of our project.

(Completed: 100%)

O2. We created a performance report that describes the performance of our machine learning network. Initially we had hoped to have enough data to be able to elaborately evaluate our network's performance in run-time conditions, but due to hardware limitations and delays in the first part of the project, this was no longer feasible. We were, however, able to report on the network's performance on training data and about the performance of this network running on a real time embedded platform.

(Completed: 100%)

O3. We demonstrated the software on an autonomous cleaning robot that we are working on at the moment. To this end, we equipped the robot with a stereovision camera and integrated our software on the Jetson TX2 computer that is the main computing unit of the cleaning robot. The demo was executed in an indoor test facility, in which the robot autonomously moved through the environment, making depth maps from the stereovision data.

(Completed: 100%)

O4. During this project, we thoroughly evaluated the IEC61508 functional safety standard and we were able to compare the requirements and recommendations of this standard with the current properties of RobMoSys. This led to the identification of points where RobMoSys could be expanded or improved to make the use of RobMoSys fit better with functional safety. This report is a public deliverable of our project.

(Completed: 100%)

Coach's perspective

The company brought in a very interesting focus on system safety, with already the state of the art implemented in a clean way. The work and interactions in RobMoSys enabled the company to broaden their scope towards future-proof safety architectures, by identifying the core architectural patterns to compose safety around the correct subset of components in their systems. The interactions with the company were extremely structured and professional.

3.2 Instrument # 2: Individual ITP Results

3.2.1 Metacontrol for ROS2 systems (MROS)

ITP Metacontrol for ROS2 systems (MROS) is conducted by the department of Cognitive Robotics at Delft University of Technology (TUDelft), the UPM-CSIC Center for Automation and Robotics at Universidad Politécnica of Madrid, the department of Signal Theory, Communications, Telematic Systems and Computation Department at University Rey Juan Carlos (URJC), Robert Bosch GmbH and the Computer Science Department at IT University of Copenhagen.

The coach assigned for this ITP was Prof. Christian Schlegel (THUlm).

Project objectives

The objective of MROS is to leverage the RobMoSys model-based approach at runtime, to provide a solution for ROS2 systems, based on architectural self-adaptation driven by ontology reasoning on the architecture models. The solution will be applied to the ROS2 Navigation stack, and demonstrated in two variants of the RobMoSys Intralogistics pilot, with the Thiago platform and with a Bosch consumer-product prototype. MROS will connect the ROS2 and RobMoSys frameworks, and further enable a metamodeling-sound usage of ontologies for robot software architecting. Its specific objectives are:

O1. Provide a reusable meta-modeling solution for reliable robot skills through architectural adaptation at runtime with clear separation of concerns for task, contingency and system handling.

O2. Demonstrate the value of the RobMoSys model-based approach for ROS2 systems, implementing the metacontrol solution for the ROS2 Navigation in an industrial pilot case.

O3. Demonstrate the value of ontologies for metamodeling and knowledge-based reasoning at runtime in the context of the RobMoSys model-driven approach to robot software

Reported results

O1: MROS has developed a ROS2 solution for reliable navigation skills in ROS2 that implements architectural adaptation through a runtime Metacontrol component. This has been released as a set of ROS2 libraries and exemplary systems the public repositories of the MROS project.

(Completed: 100%)

O2: the value of the MROS solution and RobMoSys modelling approach has been validated in two final Pilot cases, including a research scenario with the Tiago commercial research platform (URJC), and a consumer prototype of TRL 4/5 at Bosch.

(Completed: 100%)

O3: The value of ontologies for robotic modelling has been validated by using ontology engineering to develop the MROS ontologies that were the basis for the MROS Metamodel and the reasoning for the Metacontrol.

(Completed: 100%)

Coach's perspective

The contribution brings RobMoSys concepts to ROS2 by the example of the ROS2 navigation2. The RobMoSys concepts proved to be beneficial in structuring MROS.

Due to the pandemic situation, face2face meetings with deep technical discussions were not possible anymore after the ERF 2020 (March, Malaga). Virtual meetings are not a good means for deep technical discussions.

However, physical meetings would have been of great value in already exploiting within the ITP the power of combinatorics of already existing RobMoSys navigation skills and RobMoSys coordination with MROS ROS2 navigation2 skills and MROS coordination.

3.2.2 Skill composition with verified system properties (SCOPE)

ITP Skill composition with verified system properties (SCOPE) is conducted by the Department of Informatics, Bioengineering, Robotics and Systems Engineering at University of Genova and the department of Humanoid Sensing and Perception at Fondazione Istituto Italiano di Tecnologia (IIT).

The coach assigned for this ITP was Dr. Matteo Morelli (CEA).

Project objectives

The goal of the SCOPE project is to contribute to the RobMoSys ecosystem by proposing methods and tools to enable the assessment of system-wide safety properties at the behavioural level (the "deliberative layer"). With reference to the RobMoSys meta-model for robotic behavior, the goal of SCOPE is to provide tools that analyse and derive properties of a task by composing the properties that describe its skills and the environment, and, at runtime, ensure the correct execution of a task by monitoring it and propagating anomalies detected at the level of the skills.

O1. To extend the RobMoSys ecosystem with specification and modelling languages for skills and the environment that uses real-time temporal logic and allows to specify resource usage.

O2. To define operational semantics of communication patterns involved in skill orchestration, so that skill constraints can be propagated to the deliberative layer, i.e. provide tools for modelling communication between the deliberative and runtime layers.

O3. To develop tools that enable to increase confidence in the correct execution of robotic tasks both at compile and run-time with specific focus on safety properties.

O4. To integrate in the RobMoSys ecosystem and enlarge RobMoSys community.

Reported results

O1. The ITP described in detail the syntax and the semantics of the specification language for *(i)* describing the SCOPE model and *(ii)* specifying properties about the SCOPE model. In particular, the model consists of the description of a parametric behaviour tree (task plot), the implementation of the skills bound to the leaves of the behaviour tree (as finite state machines), a description of the connections between skills and components and a complete description of the blackboard entries. The properties are formulas of a suitable temporal logic that predicate on the connections among the behaviour tree, the skills, the components and the blackboard.

(Completed: 100%)

O2. SCOPE defined the syntax and semantics of behaviour trees and skills, together with *(i)* interaction patterns among skills and *(ii)* interaction patterns between skills and other components at the functional level.

(Completed: 100%)

O3. SCOPE implemented verification tools and runtime monitor generation based on a scenario where the IIT-R1 robot must fetch an object from a user and reach a predefined location ensuring that the battery level is sufficient. For design-time verification the strategy is *(i)* to translate the SCOPE model to an intermediate representation, *(ii)* to compile this representation in C language using Posix threads and *(iii)* to provide the compilation result to CPAchecker for the formal verification. Run-time monitors are extracted from the SCOPE model and implemented as software components that observe the signals exchanged by the BT, the skills and the functional components, and verify that they are consistent with the requirements.

(Completed: 100%)

O4. SCOPE designed the model according to the RobMoSys composition structures and the developed tools complement and can be used in conjunction with RobMoSys software baselines such as Papyrus for Robotics. All material is available at https://scope-robmosys.github.io/.

As outlined in the final report, a paper acknowledging SCOPE was accepted in ICRA 2020, another paper based on SCOPE research was submitted to ICRA 2021 and a journal paper is underway. In addition, a video demonstrating the SCOPE scenario has been sent to IJCAI 2021 and the results of the SCOPE project will be presented at the I-RIM 3D EXPO 2020 as part of the 2020 Rome Make Faire. SCOPE capabilities will also be demonstrated in a virtual workshop in early 2021.

(Completed: 100%)

Coach's perspective

The project progressed very well during all its duration. The SCOPE consortium attended the kick-off meeting in Munich and interacted regularly with the coach afterwards to discuss project advancements.

SCOPE identified synergies with SAFECC4Robot and ForSAMARA ITPs.

The strong technical background of the consortium allowed them to achieve all the milestones and deliverables. As former ITP participant in the first Open Call, the consortium was fast in the identification of the right RobMoSys composition structures to link with, which proven to be of great value to realize separation of roles and concerns and the bridging between existing RobMoSys-conformant baselines, such as Papyrus for Robotics, and the SCOPE toolchain for design-time verification and run-time monitoring of robot skill compositions.

The dissemination activity was satisfactory. Overall, the SCOPE contribution is very valuable from both scientific and technical point of view.

3.2.3 Formal Safety Analysis in Modular Robotic Applications (ForSAMARA)

ITP Formal Safety Analysis in Modular Robotic Applications (ForSAMARA) is conducted by the Joanneum Research Forschungsgesellschaft mbH, Technische Universität Wien and PILZ.

The coach assigned for this ITP was Dr. Huascar Espinoza (CEA).

Project objectives

ForSAMARA project focuses on formal verification of collaborative robotic applications against safety properties. In particular, the ITP applies model checking methods where safety-critical situations can be identified by the result of a mathematical verification proof during the design time. Researchers focus on the definition of rules and implications that are clearly traceable to corresponding sections of current collaborative robotics standards (such as ISO/TS 15066 and ISO 10218-1/2). Thus, specific technical safety requirements, represented in the level of detail of the model, are checked. This check covers application specific critical scenarios identified in risk-analysis processes. This goal will be accomplished by following the specific objectives named below.

O1. Conduct a comprehensive safety analysis in industrial collaborative robotic applications.

O2. Conduct a design-time verification of composable component models and models describing the robot's environment defined within the RobMoSys ecosystem.

O3. Utilize formal verification methodologies for proving safety properties on the defined system model.

O4. Abstract component and architecture models to handle complexity issues and achieve a well-defined balance between expressiveness and verifiability of models.

O5. Model an adaptive, modifiable behaviour within the component models or given by architectural variabilities.

O6. Conduct a functional verification even under the presence of previously defined and modeled adaptive behaviour.

O7. Define a methodology (process) for guided formulation of machine-readable safety properties derived from application-relevant standards.

O7. Define and test a methodology (process) for guided formulation of machine-readable safety properties derived from current collaborative robotics standards (such as ISO/TS 15066 and ISO 10218-1/2)

Reported results

O1. ForSAMARA collaborated with its industrial partner Pilz GmbH, who conducted a professional risk assessment for the ForSAMARA use case Demonstrator 1 (D1). At their conclusion, they proposed three steps in a safety perspective that have to be evaluated on the use case implementation. Steps are related to the application of inherent safe design, limitation of movement speeds and force thresholds and organizational measures.

(Completed: 100%)

O2. The representation of an environment model is at the time of the report not possible in the Papyrus for Robotics tool. Thus, ForSAMARA chose the static environment OctoMap/voxel model, an external software application commonly used in the ROS community. Demonstrator 2 will investigate further extensions of D1 related to environment and modeling variables.

(Completed: 100%)

O3. ForSAMARA implemented system safety requirements as properties of the demonstrator D1 which models a simple collaborative assembly use case with two different robot setups, a Panda robot from Franka Emika and a UR5 robot from Universal Robots. Both implementations strictly fulfill the requirements and conceptually represent equal models in the RobMoSys context. The implementation of the use case is independent of a specific robot model as well as its specific physical setup. However, a set of basic requirements has to be fulfilled by the robot system. These requirements are reflected by a set of robotic skills implemented in component models specified in the RobMoSys toolchain.

The task of D1 includes basic linear movements, where the robot serves as an assistive system and supplier of work pieces. The safety properties, which are checked on the SMV model, have to be defined. In D1, ForSAMARA first presented potential unexpected behavior between the robot and a specific part of the human body in a simplified way. In the following expansion stages, the analysis options are significantly expanded, supported by suitable, more detailed modeling. This will allow a verification against a large number of dangerous situations and risks.

(Completed: 100%)

O4. The ITP modeled and designed D1 using Papyrus for Robotics toolchain to describe the behavior of the robot and the associated workflows as well as the significant characteristics of the robot system. Specific parts of the models defined in Papyrus were then transferred into abstracted artefacts used for model checking. As a model checking tool we selected nuXmv, where behavioural models of the system under verification are specified using state space representations. The properties to be checked are formulated in a Propositional Temporal Logic (PTL). They still have to formulate PTL properties specifically reflecting system safety requirements.

(Completed: 100%)

O5. The approach for implementing the concept of demonstrator D1 is based on a series of submodules (puzzle parts). For the definition of workflows and the behavior of the robot, they used modeling capabilities of the Papyrus for Robotics tool as well as nuXmv specific schemes represented by state machines. The robot's environment for demonstrator D1 is defined by an octomap/voxel representation. Such a representation is very efficient for simulation-based and runtime modeling of environmental conditions but challenging to abstract for model checking processes. They integrated their approach into the RobMoSys ecosystem by utilizing its possibilities for modeling dynamic behaviours of robots. More precisely, they used the behavioural tree to model the behavior and translate it to source code for the model-checking tool nuXmv.

(Completed: 100%)

O6. The first model checking runs were performed after the complete setup of all tools and the creation of the required modelling components of demonstrator D1. The verifiable model of D1 consists of eight source files which are merged under given dependencies. The model checker is called with the help of a SMV script which makes the operation of the software on command line level much easier. Demonstrator 1 is considered a simplified model that shows the essential concept of the corresponding modeling and model transformations as well as the steps towards formal verification. This approach is feasible and first results in terms of collision detection between an object of the environment and the moving robot have been shown. However, formal verification involves several challenges which researchers plan to solve by clear modelling approaches and a corresponding abstraction.

(Completed: 100%)

O7. During the project two different approaches in order to represent safety properties were developed. First, application specific properties as defined by requirement documents in natural language can be translated to machine readable, and verifiable formulas. However, this process has to be executed manually or semi-automated. The ForSAMARA team focuses on the second approach which is described as follows. For the representation of mechanical clamping hazards as occurring at collaborative robot applications a graphical modelling approach is introduced. Hazardous zones as identified in the system's risk assessment are defined as hazard-boxes in the environment model. This graphical representation is more easy to use and understandable for the verification engineer in contrast to specifying property formulas in temporal logic. Nevertheless, for the developed UR5 demonstrator this hazard modeling process has been applied according to a risk assessment done by the project partner PILZ.

(Completed: 100%)

Coach's perspective

ForSAMARA has performed an excellent job from the kick-off meeting. It was involved in discussions with other ITPs working on safety-related issues and has shown a good degree of complementarity with SCOPE and SafeCC4Robots. This ITP follows a well-structured and systematic work. They defined sprints to first establish the conceptual framework and then to develop the two versions of case study prototype and demonstrators. All the partners were well integrated in the projects and provide complementary perspectives to the progress (conceptual, implementation and industrial application). From the technical point of view, ForSAMARA has well integrated RobMoSys concepts and technology (in particular Papyrus for Robotics) and has defined a workflow generic enough to be applied to different problems in robotics.

ForSamara defined a first demonstrator (D1) as well as the risk assessment methodology in conformance to safety standards. In particular, the ITP integrated the principles of modularity, human-robot collaboration as well as environment modelling as part of the proposed formal verification process. They also integrated model checking tools used for formal verification of safety properties. The ITP prepared a video of the progress and presented at ERF 2020.

Demonstrator 2 (D2) extended demonstrator 1 (D1) by adding significant functions regarding a collaborative interpretation of the defined use case The model checking specifications have also been extended in demonstrator D2: e.g; it is now possible to also check against collisions that occur dynamically, such as a collision with a specific human body region. In D2, we realised synchronisation meetings with CEA (coach organisation) to integrate ForSAMARA developments in Papyrus for Robotics, including: 1) tooling workflow definition, 2) risk assessment extensions to Papyrus for Robotics, and 3) modelling of assertions for model checking. To represent and verify models that include variability, some core features of the NuXmv model checker were used. However, the model state space of the implementation increases accordingly, which has a significant influence on the processing time of the model checker. This has to be considered when using model checking techniques.

ForSamara did an excellent work, which is relevant for human-robot interaction and safety assessment. For a further consideration of collision situations, other robot parts such as joints can also be considered in addition to the TCP position. The performance of the modeling could still be improved to facilitate model checking with NuXmv. The defined workflow will be extended with increased semi-automatic and automatic model generation.

In addition, ForSamara did a good work in dissemination. They presented their work in conferences and virtual exhibitions (IPAS 2020, SIT 2020, ERF 2020) and submitted applications for Austrian awards in robotics research (JOANNEUM RESEARCH Best Performance Award and Austrian Standards: Living Standards Award; still waiting for a response). They were very active in LinkedIn and Twitter.

3.2.4 Component Composition from Real-time Function Blocks (COCORF)

ITP Component Composition from Real-time Function Blocks (COCORF) is conducted by Dr. Markus

Klotzbuecher.

The coach assigned for this ITP was Dr. Enea Scioni (KUL) from M01-M04, and Dr. Marco Frigerio (KUL) from M05 on.

Project objectives

COCORF aims at combining the RobMoSys composition approach with the proven microblx real-time function block framework. This will enable building modular and predictable control systems by composing reusable blocks based on the function block model of computation. Microblx introspection facilities will be used to generate RobMoSys compliant digital function block data-sheets containing formalized constraints and QoS properties. Interoperability to existing robotics middlewares as well as separation of criticality domains will be realized by means of mixed-port function blocks. To validate the technology and to support early adopters, the approach will be illustrated by developing a generic control reference architecture and accompanying demonstration for the Assistive Mobile Manipulation Pilot. This project has three main specific objectives:

O1. Achieve that the composition DSL and the digital data-sheet for function blocks become RobMoSys compliant.

O2. Make available Mixed-port function blocks to ROS.

O3. Complete a generic control reference architecture.

O4. Disseminate results at relevant robotics community events.

Reported results

O1. COCORF implemented the DSL composition extensions and the digital data-sheet for function blocks developed during this reporting period. A new didactical example was prepared and documented in the microblx repository. The composition extensions were implemented, tested and documented and a first version has been announced on the RobMoSys discourse platform. Block digital data-sheet support extensions were implemented in the ubx-modinfo tool, which is being used to auto generate online digital data-sheets for the core function blocks. The code changes for both features have been merged to the microblx development branch and have been released as part of the 0.8 (https://github.com/kmarkus/microblx/blob/v0.8.0/ChangeLog.md) release.

(Completed: 100%)

O2.Mixed port function blocks for ROS have been developed (https://github.com/kmarkus/microblx-ros) and publically released (https://discourse.ros.org/t/announcing-the-microblx-ros-connector-block/14691) under the BSD license. The block supports the common standard types, as well as KDL based geometry types. Importantly, it can be safely deployed in a hard real-time context without impeding real-performance.

(Completed: 100%)

O3. The example of a cascaded control system is used as a reference architecture to illustrate the composition DSL extensions. Due to travel impediments, the demonstration for the Assistive Mobile Manipulation Pilot was (with approval of the COCORF coaches) substituted with a simulated demo which is based on collaborative, development efforts with the VeriComp ITP. This demo illustrates the power of the microblx composition DSL by building several variants of an application using composable and reusable models.

(Completed: 100%)

O4. Dissemination has taken place in form of a screencast shown at the European Robotics Forum, where the composition extensions have been introduced. Further dissemination took place in form of an invited talk at the ROS2 real-time working group. There have been a small number of private email exchanges with potential future users from industry.

(Completion: 90%; but will continue beyond the end of COCORF.)

Coach's perspective

The progress of the COCORF ITP was always smooth and steady, both from the technical point of view and management, as well as in relation with coaching.

The internal milestones of the project due in the different periods (technical developments and deliverables) were successfully achieved. The results are documented in the main source code repository of the project (publicly hosted on the web), as well as in the relative RobMoSys wiki page; furthermore, intermediate results were also presented at the European Robotics Forum (ERF).

The ITP team successfully attended the ITP kick-off meeting in Munich, participating actively and investigating synergies with respect to: (i) other ITPs and (ii) industrial RobMoSys-core member partners regarding the development of the demonstrator case. Synergies were detected with the CMCI and VeriComp projects, and led to some integration between the software products developed within the different projects.

Overall COCORF (and the microblx framework, which is the technical subject of the project) stands as valuable complement of the RobMoSys baseline tooling, by providing a component-based framework for the development of real time systems. The framework exposes and conforms to some of the RobMoSys principles and best practices, such as composability, separation of concerns, availability of data-sheets, interoperability with other middleware.

3.2.5 Verifiable Composition of Dynamics and Control Algorithms for Robot Motion (VeriComp)

ITP Verifiable Composition of Dynamics and Control Algorithms for Robot Motion (VeriComp) is conducted by the Research Institute for Cognition and Robotics (CoR-Lab) at Bielefeld University and the Bonn-Rhein-Sieg University of Applied Sciences (BRSU).

The coach assigned for this ITP is Dr. Herman Bruyninckx (KUL).

Project objectives

VeriComp aims at addressing one of the identified ecosystem challenges of RobMoSys by augmenting functional composition inside components with verifiable properties and domain-specific extensions, linking the latter to the system-level RobMoSys composition structures. Particularly, VeriComp objectives are:

O1. Develop Meta-models for verifiable motion control algorithms.

O2. Verify composable motion control algorithms.

O3. Summarize video for the robotics community.

O4. Create Technical user stories in VeriComp.

Reported results

O1. VeriComp collected requirements and coarse domain models in the context of a use-case in which a dual-arm robotic system is tasked to cooperatively lift a box. This robot control scenario helped select verifiable properties and identify constraints that must be checked by the tooling to assure a correct composition.

From the identified requirements and domain models, they derived four formal meta-models for verifiable motion control algorithms, namely Algorithms; Geometry, mechanics and kinematic chains; Control and

Timing meta-models. Additionally, three meta-models, that are not VeriComp-specific, were included in the list of domain models because they provide the necessary framework for working with verifiable properties in the context of a robotic system, namely Composition Structures; Block-Port-Connector and Components.

Meta-models treatment was restricted to their structural aspects and static semantics. And metamodels structure was addressed by expressing metamodel elements and their relations graphically using the UML inspired graphical notation provided by the Ecore framework.

On a technical level, VeriComp meta-models were embedded in domain-specific languages developed in the Meta Programming System (MPS), an open-source language workbench developed by JetBrains. VeriComp uses MPS as a realization alternative for working with models and meta-models which conform to the RobMoSys composition structures. For researchers, the underlying technology of MPS is a labeled property graph that conforms to the RobMoSys Entity-Relation Model.

Therefore, VeriComp re-expressed the mechanisms for static semantics and structural constraints identified with the use case within the MPS language workbench.

The purpose of meta-models is the automatic creation of executable artifacts. VeriComp selected a four-stage approach to create executable artifacts: (meta) modelling, model transformation, code generation and compilation. These stages bridge three levels of abstraction (the conceptual level, the level of abstract data types which adds choices with respect to the representation of information and finally meta-models and models for digital data and data types) where domain concepts appear.

These meta-models are RobMoSys conformant since they are related to the Component Definition RobMoSys metamodel, the Performance meta-model as well as the System Component Architecture RobMoSys meta-model.

In parallel to the model transformations for generating artifacts at the digital data representation level, VeriComp plans to extract and transform information from the meta-models covering the conceptual level to generate digital data sheets alongside the source code and executable artifacts.

(Completed: 100%)

O2. Regarding composability and verification of motion control algorithms, the work in VeriComp led to the implementation of model-based checks of constraints for domain-specific concepts that can be evaluated at design time to provide early feedback. Exemplary design time checks are available in the language modules for the Algorithm, Control and Kinematics domain, e.g., to enforce the compatibility of types and specific semantics in algorithm models. Further, we explored possibilities to transform the modeled algorithms into an intermediate model representing control flow graphs with static single assignment semantics to uncover data-flow and scheduling problems.

Besides, the developed meta-models in O1 paved the way to define the *Nanoblocks* pattern which enables function developers to structure the state of lightweight components in a way that makes it easy to fully expose that state to the level of algorithm models. The pattern allows to declare data flowing in and out of the component as well as storage for intermediate results which are represented uniformly as pointers to storage that the execution environment sets up and connects to one or more *Nanoblock*-based lightweight components. The *Nanoblocks* pattern also facilitates packaging certain lightweight components as proper components for a particular component-based execution environment. To this end, VeriComp developed several composable function libraries such as *dyn2b* for the geometry and dynamics domains, *ctrl2b* for the control domain, *robif2b* for dynamics and collision simulation of a robot, and *vis2b* for 3D visualization of kinematic chains.

(Completed: 100%)

O3. The summarizing video of the VeriComp ITP reports on the application of the modeling approach along a running example that specifies a dynamics solver algorithm using the language modules developed in VeriComp such as the Algorithm, Control and Kinematics DSLs and its mapping towards the microblx real-time component framework. The demonstration video is published at the project website available at: https://rosym-project.github.io/vericomp

(Completed: 100%)

O4. VeriComp demonstrated and evaluated the formalized meta-models, verifiable properties for composable specification of algorithms as well as the modelling environment that employs these meta-models through several RobMoSys technical user stories.

The technical user stories cover the complete development life-cycle of robot motion applications and includes, to name a few, the design, composition, verification and deployment of robot control algorithms. To this end, VeriComp supports important RobMoSys ecosystem roles such as the *function developer*, *component supplier* and *performance designer* in their development activities.

The VeriComp modelling approach enables function developers:

- to employ domain-specific abstractions in order to design composable motion-specific operations from relevant domains such as kinematics, dynamics and control,
- to model and develop algorithms in an incremental and iterative manner supported by a thorough application of both composability and compositionality principles,
- to propagate any semantic violations back to the function developer by relying on the composability and compositionality principle paves the way to develop correct-by-construction algorithms and composition of algorithms,
- to retrieve feedback for invalid data and control flows within algorithm compositions,
- to automatically synthesize semantically correct domain-specific algorithms,
- to compose and interleave control flows in order to reuse computational state of algorithms,
- Even though VeriComp primarily addressed the function developer the component supplier is supported as well. To this end, the VeriComp modelling approach enables component suppliers:
- to package algorithms as components so that they are reusable by other developers
- to generate a data sheet for a component in order to establish a link to the RobMoSys ecosystem.

(Completed: 100%)

Coach's perspective

The consortium organised the interactions with the coach very efficiently, and monthly basis, and with a clear focused agenda for every meeting. In this way, progress was very clear, and incremental. The biggest result is to have been able to bring more explicit structure in the thinking and designs of the consortium team; that team was very junior, very eager to make progress, but struggled with the larger system scope and impact of their developments. It is exactly this one-on-one dissemination of the RobMoSys' approach that makes the ITP concept so valuable. The negative side is the high human effort involved; but this is obviously not caused by the consortium, but by the inherent complexity of robotic system design and implementation. It was very nice to see that towards the end of the project, and effective and constructive cooperation has taken place between VeriComp and CoCorf.

3.2.6 Energy-Guided Control Stacks and Robot-Software Architectures using Model-Driven Design (RobMoSys-EGCS)

ITP Energy-Guided Control Stacks and Robot-Software Architectures using Model-Driven Design (RobMoSys-EGCS) is conducted by the department of Electrical Engineering, Mathematics and Computer Science at University of Twente, the Netherlands Organisation for applied scientific research (TNO) and VIRO.

The coach assigned for this ITP was Dr. Herman Bruyninckx (KUL).

Project objectives

EGCS aims to develop a methodology to further extend the concepts of "energy-awareness" to the whole motion stack, such that all higher abstraction levels in the motion stack are equipped with means to realize the physical interactions of the system with the environment. In practice this actually means using the energetic information of the components as status information of those components, such that higher-level control layers can infer whether the system operates as expected. By doing so, system-level aspects like safety, fault diagnosis / monitoring can benefit, contributing to the autonomy and reliability of the system. This "energy-awareness" provides functionality at the Supervisory and Sequence Control level for (1) dealing with timing differences, buffer effects; (2) dealing with fault detection and fault handling; (3) predicting system-state properties ensuring larger stability areas. Relating to the RobMoSys levels of abstraction, our motion stack enables composability, reusability and replaceability, preserving separation of levels. The work will be tested / validated by industrial-level case studies as well. EGCS has the following specific objectives.

O1. Develop EGCS meta models through the extension of the IPC meta-model and the Research test case, focusing on higher levels of the control stack.

O2. Define industrial-linked use cases and implement them in the role of system developer, including description of user stories for that role.

O3. Evaluating the usability of the extended IPC meta model.

O4. Ensure the robotics community, in particular the RobMoSys community, is aware of the efforts, potentials and results of the project.

Reported results

O1. EGCS control stack meta-models have been developed, see Deliverable 2.2. These meta-models aim to extend the existing meta-models developed in the previous ITP (EG-IPC). And generic control stacks have been formulated.

(Completed: 100%)

O2. EGCS has defined industrial use cases with two end-user partners, TNO and VIRO. The TNO case, an underwater robot case, mimicked by using two Franka robot arms, one being the underwater robot, and the other emulating the relative movement between the underwater robot and the underwater object to be manipulated. The VIRO case is an energy-aware generic robot control development framework and is applied to develop code for robotic systems in the context of human-robot interaction. Both cases have been designed, built and tested. and used as a test vehicle for the meta models, see O3.

(Completed: 100%)

O3. During the tests as indicated at O2, the usability of the extended IPC meta model has been tested. It appeared to give valuable structure and support to increase modularity and clarity of the software. It also reveals that proper accommodation of real-time demands is not yet covered by the RobMoSys tooling, and thus careful choosing how to map the energy-aware software architecture onto RobMoSys components is important.

(Completed: 100%)

O4. Results of EGCS have been discussed with the Coach, and as such we expect that these reach the RobMoSys core team. The first scientific paper is about to be submitted to the ROSE21 conference. The public wiki page is still to be made, using the existing technical documentation. Our GIT repository is still to be finalized (i.e. add some documentation and review structure) before we will make it public.

(Completed: 50%)

Coach's perspective

This consortium provides us with an ideal mix of stakeholders, from junior developers to tech transfer organisation and a real company. The interactions have been very regular, and very well prepared. The outcome of this project is tremendous, taking into account the huge complexity of the real-time system-to-system interactions that the consortium was trying to tackle. The result is not yet a fully implemented set of functionalities and tools, but that was too ambitious to expect from the outset anyway. The major added value is the very structured modelling of the large-scale system components, with clear responsibilities and composable modules.

3.2.7 Safety Component Composition for Robots (SafeCC4Robot)

ITP Safety Component Composition for Robots (SafeCC4Robot) is conducted by Tecnalia.

The coach assigned for this ITP was Dr. Ansgar Radermacher (CEA).

Project objectives

The SafeCC4Robot project aims to develop a methodology and tool support for the creation of robotics components ensuring safety at system level. This support targets to enable the usage of robotic component suppliers in different systems and to remain systems safe after composition. This goal will be addressed as follows.

O1. Specify the requirements and conceptual approach, as well as design and implement the SafeCC4Robots tools interfaces and adaptation of existing tools.

O2. Specify the benchmarking plan by using a case study of the robotics pilots, as well as perform testing and develop a demonstrator to benchmark the project goals.

O3. Plan, coordinate, and control the activity in the project in order to ensure that its technical objectives are reached and there is an adequate level of dissemination and communication.

Reported results

O1. SafeCC4Robots presented a conceptual approach for the SafeCC4Robot project and the high-level requirements. Terms and definitions are specified. 16 high level requirements of the SafeCC4Robots platform from the user point of view were discussed and defined. They are organized based on Requirements for system modelling and composability and requirements for defining the Safety Dashboard of RobMoSys at component level. All functionalities have been implemented in Papyrus4Robot platform and are integrated and available for RobMoSys users.

(Completed: 100%)

O2. The ITP provided twelve metrics that quantify the extent to which SafeCC4Robot's methodology and tools support the creation of components for robotics that ensure safety at system level during and after the composition. A real robotic device developed by TECNALIA HEALTH, ArmAssist, has been modelled and implemented using RobMoSys core functionalities and, its safety conditions defined using SafeCC4Robot modules. The case study has been assessed against the metrics defined with successful results and ArmAssist has been successfully validated with SafeCC4Robot modules. Two papers describing this case study will be send to congresses at the beginning of next year, once the project is over (the date of the congresses were postponed due to the COVID situation).

(Completed: 100%)

O3. The dissemination plan and objectives stated at the beginning of the project have been largely achieved (e.g. number of followers, tweets, impression rate). The events attendance was high at the beginning of the project, which help to get many followers of the project on board, even if most of in-situ events for year 2020 were cancelled. In any case, the project on-site events were substituted by on-line meetings. In relation to the exploitation plan, it has been stated by the end of the project but some of the activities to carry out the exploitation plan were undertaken within the scope of the project (e.g. selection of ArmAssist case study for the pilot – it will facilitate the exploitation of SafeCC4Robot solution in the groups that develop robots within TECNALIA).

(Completed: 100%)

Coach's perspective

The project developed the possibility to define properties, assertions and contracts. The specification of these is done in the form of a UML profile. In order to improve usability, additional components have been developed to improve Papyrus for Robotics. These include property view extensions with tables. Since this work required in-depth knowledge of Papyrus-for-Robotics it was partially supported by the CEA.

The results are highly useful for using Papyrus-for-Robotics in safety relevant projects, including the ITPs SCOPE and forSamara.

Project dissemination was satisfying, for instance two leaflets have been prepared and distributed in the relevant events. The ITP also created a Twitter account for the project (@S4robot) to which main achievements have been tweeted. Deliverables and project reports have been uploaded to the

RobMoSys ITP Monitoring Platform. The ITP attended academic conferences for project dissemination.

3.2.8 Composable Models for Compliant Interaction Control (CMCI)

ITP Composable Models for Compliant Interaction Control (CMCI) is conducted by the Research Institute for Cognition and Robotics (CoR-Lab) at Bielefeld University (UniBi) and the Institute for robotics and process control (IRP) at Technische Universität Braunschweig (TUBS).

The coach assigned for this ITP was Dr. Enea Scioni (KUL) from M01-M04, and Dr. Herman Bruyninckx (KUL) from M05 on.

Project objectives

The overall ambition of the CMCI project is to provide composable domain-specific models, model transformations and a controller implementation together with tools to easily synthesize the required motion control components for advanced robots from a model-driven specification of compliant interaction tasks. Specifically, the project will:

O1. Provide models and a meta-model for the description of prioritized control tasks with specific constraints for compliant motion control.

O2. Use a software library to implement a hierarchical Stack-of-Tasks (SoT) approach that can evoke solvers for sequential quadratic programming (SQP) and that can be configured using models through a dedicated configuration language.

O3. Provide a comparably easy-to-use model-based specification language for compliant interaction tasks.

O4. Provide a model-to-model transformation that generates the SoT-configurations, which in turn configure the SQP solver from the compliant interaction models.

Reported results

O1. This deliverable reports on the conformance of extended language modules for compliant interaction in the Compliant Simulation and Modeling Architecture (CoSiMA) environment. CMCI adapted the already existing language modules of CoSiMA to conform to RobMoSys. This includes a domain analysis for Compliant Interactions (CI) related to (i) suitable control strategies that can be applied for CI situations and (ii) Contacts, Contact Situations, and Contact Constraints enforced by specific types of contacts.

CMCI modeled two stacks of tasks for humanoids applications performing manipulations based on the domain analysis. To achieve a specific behaviour, control tasks require prioritization, they made explicit the prioritization structure to specify different robotic behaviours and interactions with the environment. And they developed the CI Behavior Description language module to describe the interaction with the environment in a geometry-based manner.

(Completed: 100%)

O2. CMCI created a control task prioritization framework that makes use of a QP-based solver implementation as described. This C++ framework is capable of realizing (high-level) task-specific SoT configurations that can be executed in real-time. Further, local as well as global constraints are supported, which may artificially constrain the acceleration, force, and joint positions. It can be configured in a model-based approach and provides a data sheet that describes functional and non-functional properties as well as additional information for a system-level integration.

(Completed: 100%)

O3. In the course of CMCI, mainly two domain-specific languages were developed. Both languages can be used independent of each other, but can also be combined with other existing languages of CoSiMA (see O1). The first language is concerned with enabling a compliant interaction behavior description (see O1). Whereas, the second language targets the solver-agnostic modeling of Stack-of-Tasks, including models with explicit requirements for the supported solver backends. Both languages are implemented using the open-source language workbench MPS from JetBrains.

(Completed: 100%)

O4. A model transformation was developed that transforms a Stack-of-Tasks model into a C++-based configuration for the targeted solver API. To this end, the transformation also takes the explicitly modeled information regarding the chosen solver backend (see O3) into account. The result of the case study of CMCI was a fully generated and executable OROCOS RTT component, wrapping around the configured OpenSoT API, that realizes the behavior conforming to the modeled tasks.

Additionally, initial insights were gained on the mapping between the common constraints of the CI Behavior Description language and the individual tasks of the SoT language. However, this appears to be a topic that requires further investigation.

(Completed: 100%)

Coach's perspective

This consortium brought a very young and super-enthusiastic team into the project. So, there was a high impact of transmitting core RobMoSys structures and approaches towards this team, but the concrete outcome that they might have dreamt about was, obviously, not realised to the full extent; this is certainly not due to a lack of efforts or interest from the consortium, but to the inherent complexity of the problem it wanted to solve. The dissemination results, however partial, are still worth the project funding, by far.

3.2.9 QoS Metrics-In-the-loop for better Robot Navigation (MIRoN)

ITP QoS Metrics-In-the-loop for better Robot Navigation (MIRoN) is conducted by The Computer Systems and Telematics Engineering department at University of Extremadura (UEX), the department of Computer Languages and Sciences at University of Malaga and Blue Ocean Robotics (BOR).

The coach assigned for this ITP was Dr. Dennis Stampfer (THUlm).

Project objectives

MIRoN aims at contributing to RobMoSys with a model-based framework for dealing with predictive, adaptive robot navigation. This adaptation will be based on the systematic use of models for dynamically reconfiguring the robot behavior, defined in terms of Behaviours Trees, according to the runtime prediction and estimation of Robot Quality-of-Service (QoS) metrics defined on non-functional properties. Validation of the proposal will be addressed in the two industrial pilots related with navigation, offered by RobMoSys: the THUlm Intralogistics Industry 4.0 Robot Fleet pilot and the PAL Assistive Mobile Manipulation Pilot.

This goal is developed via the following specific objectives.

O1. Develop the MIRoN framework.

O2. Develop MIRoN demonstrators.

O3. Disseminate results.

Reported results

O1. The MIRoN framework has been developed as a set of Eclipse plug-ins and additional software tools, ready to be installed within and executed together with the SmartMDSD Toolchain: one of the RobMoSys baseline frameworks for robotics software development. The MIRoN frameworks provides designers with: (i) a textual model editor allowing them to specify variation points in the robot behaviour and define how these variation points should be configured at runtime according to the perceived situation; and (ii) a model-to-code generator that, taking the previous models as an input, generates and appropriately configures the runtime infrastructure needed to monitor relevant non-functional properties and, according to their evolution, perform the appropriate behaviour adaptations to meet the required robot quality-of-service.

(Completed: 100%)

O2. The MIRoN framework has been validated through the development of two demonstrators: one in the context of an intralogistics scenario (similar to the Intralogistics Industry 4.0 Robot Fleet pilot provided by the Ulm University of Applied Sciences) and another one in the context of a healthcare scenario (similar to the Assistive Mobile Manipulation Pilot provided by PAL Robotics). Both demonstrators have been developed on a simulated environment, as the COVID-19 pandemic has made it impossible (due to travel limitations) to perform the validation on the real-world pilots provided by the RobMoSys partners. The models developed for both scenarios, together with some explicative videos are publicly available in a dedicated GitHub repository.

(Completed: 100%)

O3. Regarding result dissemination, it is worth noting that all the project outcomes (modelling tools, generators, demonstrator models and videos, publications, presentations, installation and usage instructions, etc.) are publicly available in a dedicated GitHub repository. Besides, the MIRoN team has punctually reported progress in Twitter, publishing more than 30 tweets and obtaining more than 300 followers during the project lifetime. The project outcomes have been presented during the RobMoSys meetings and also in the context of two workshops: the 2020 Ibero-American Meeting on Variability, Configurable Systems and Product Lines, and the 21st International Workshop of Physical Agents (WAF 2020). The work presented at WAF has been included in the Springer's book series on Advances in Intelligent Systems and Computing and has been selected to be extended and considered for publication in a special issue of the Multimedia Tools and Applications Magazine.

(Completed: 100%)

Coach's perspective

The project consisted of very experienced people who had previously participated in RobMoSys ITPs. The first use-case was defined right at the kick-off meeting and consequently developed. The project was a very good contribution to RobMoSys from a scientific / methodological point of view. There was quite a lot of dissemination, which is very valuable for spreading RobMoSys.

3.3 Instrument # 3: Individual ITP Results

3.3.1 Guidelines for Improving SmartMDSD with DDS and QoS attributes for communications (SmartDDS)

ITP Guidelines for Improving SmartMDSD with DDS and QoS attributes for communications (SmartDDS) is conducted by the department of Computer Languages and Sciences at University of Malaga.

The coach assigned for this ITP was Dr. Alex Lotz (THUlm).

Project objectives

The main goal of SmartDDS ITP is the inclusion of part of the DDS core standards in SmartMDSD, which is expected to have a huge impact on people and companies working on distributed and real-time systems, directly related to robotics, such as the IoT field, on people and companies using legacy code and algorithms that were implemented in a programming language different from C++, which want to include them in a RobMoSys component ecosystem. And as a side effect, but one of real importance is the possible compatibility with the ROS 2 existing baseline. Specifically, Smart DDS will:

O1. Discuss the guidelines for the definition of QoS attributes and DDS in SmartMDSD for communications in RobMoSys.

O2. Evaluate and discuss the prototype code generators already existing at HSU regarding communication patterns to be able to use DDS as an alternative middleware.

O3. Explore ways to help SmartMDSD developers with the design of a zero-copy gateway approach for exchanging data with already existing third-party DDS-based systems such as those developed in ROS 2 or systems which are part of smart environments using Internet of Things (IoT) devices.

Reported results

O1. The SmartSoft/DDS implementation is now fully tested and is planned to be available with the next release of SmartMDSD. It is worth noting that RTI Connext DDS licenses must be obtained before using this extension. However, the SmartSoft component developer API has demonstrated to be flexible enough to allow the inclusion of other DDS vendor implementations. Some of them have different licensing policies such as eProsima FastRTPS DDS (Apache License 2.0).

(Completed: 100%)

O2. The SmartSoft component developer API provides the interfaces to develop the communication patterns described in the corresponding RobMoSys metamodel: push, event, query and send. Ulm University of Applied Sciences (HSU) has provided a reference implementation of the patterns using the RTI Connext DDS product. With this starting point, SmartDDS have been working on: i) testing the existing codebase and, ii) working on the integration of the DDS-compliant RobMoSys components with other software ecosystems using DDS, and exploring how to connect them when different security and quality-of-service attributes do not match.

(Completed: 100%)

O3. With DDS included as an alternative middleware in SmartMDSD, the integration with other DDS-based software ecosystems is easier. For instance, the ITP is rigorously testing the SOSS (system of systems synthesizer) proposal, which acts as a common gateway between closely related middleware (https://github.com/osrf/soss). SOSS is now able to connect DDS publishers and subscribers that run in different domains and security contexts, just using a configuration file with the specific attributes of the gateway behavior.

(Completed: 100%)

Coach's perspective

The ITP coordinator fulfills the role of an advisor with respect to DDS in a very valuable way. As an expert, he confirmed the approach of how THU has structured the mapping of the RobMoSys communication patterns onto the RTI DDS implementation. This is a very valuable feedback for RobMoSys. All deliverables have been finished in time within the project period. Before Corona shutdown, two valuable face-to-face meetings took place, one in Granada and another in Malaga.

3.3.2 Guidelines for Improving Papyrus for Robotics with PDDL Planning: Planning4Papyrus (Planning4Papyrus)

ITP Guidelines for Improving Papyrus for Robotics with PDDL Planning: Planning4Papyrus (Planning4Papyrus) is conducted by the department of Signal Theory, Communications, Telematic Systems and Computation Department at Universidad Rey Juan Carlos.

The coach assigned to this ITP was Dr. Matteo Morelli (CEA).

Project objectives

This ITP aims to advise the designers and developers of the Commissariat à l'Énergie Atomique(CEA) to include a PDDL-based planner in the Papyrus for Robotics tool. This planner will be applied in the Human-Robot Collaboration for Assembly pilot, which CEA is carrying out. The objectives are:

O1. Discuss the possibilities and advantages of planning in the development of robotic behaviours.

O2. Extend Papyrus for Robotics with a PDDL-based Planner.

O3. Explore the possibility of combining a PDDL-based Planner with Behavior Trees.

O4. Analyse how to generate problems and domains in PDDL dynamically to form models that comply with RobMoSys.

Reported results

O1. During the project, Planning4Papyrus had periodic telematic meetings to discuss the advantages of using planning to develop components for autonomous robots and the integration of a planning system by PDDL in Papyrus for Robotics. During the last two months of the project, the ITP was at CEA List, in Paris, as part of a research stay related to this project. During this period, the ITP had the opportunity to push forward these discussions about design and development approaches.

(Completed: 100%)

O2. One of the design decisions made during the project was for Papyrus for Robotics to use ROS2 Planning System (PlanSys2 for short) as a PDDL-based planning system. PlanSys2 provides a PDDL plan generation and execution system that can be used by Papyrus. Planning4Papyrus had included modifications in PlanSys2 during the project to push forward this integration.

(Completed: 100%)

O3. During the first month of the stay of Planning4Papyrus in the CEA list, modifications in PlanSys2 to transform the plans in PDDL into Behavior Trees for execution were made. As a result of this work, it was written a paper for the AAMAS congress with the algorithm developed. Furthermore, Planning4Papyrus developed a mechanism to implement the actions in PDDL as Behavior Trees.

(Completed: 100%)

O4. Planning4Papyrus used RobMoSys principles in the design decisions during the meetings, both virtual and face-to-face, on the CEA list. It was clearly defined how Planning4Papyrus development complies with RobMoSys.

(Completed: 100%)

Coach's perspective

The ITP progressed very well over all the project duration. The ITP and the coach worked closely to discuss the definition of the proper mapping of automated planning components and the RobMoSys concepts, layers and structures. Finally, the collaboration brought benefits on both sides. On the one hand, the ITP refined the design of PlanSys2 to achieve better composability and reconfigurability. On the other hand, RobMoSys benefited from the ITP advices on which information should be modeled to generate problems and domains in PDDL dynamically, and at which level of abstraction this information should be modeled to hide the complexity of PDDL definitions and raise the abstraction of planning specifications.

The dissemination activity was satisfactory. Overall, the Planning4Papyrus contribution was a very valuable one.

3.3.3 Human-Robot Coproduction (HRC)

ITP Human-Robot Coproduction (HRC) is conducted by the department of Design Engineering at Delft University of Technology - Industrial Design (TUD-IDE).

The coach assigned to this ITP was Dr. Selma Kchir (CEA).

Project objectives

HRC targets to discuss with the consortium, how the RobMoSys' goal of a composable system can be integrated with a model of ensured usability by the human operator, and if possible, contribute to this. This goal will be accomplished by benchmarking the projects funded by RobMoSys with respect to the achieved usability and integration in the socio-technical ecosystem. How easy is it to adapt the system to a specific use case? How easy is it to handle the system? Is it intuitive to interact with the system? Can the real application benefit from a higher efficiency and effectiveness of the task with the help of the system?

O1. Understand better the RobMoSys approach, its current challenges and especially to develop an integrated view on the human-factors approaches that are currently applied (or not applied). To accomplish this goal, several meetings with partners and ITPs of Instrument # 1 and # 3 are expected to take place.

O2. Attend and organize workshops to present the results.

Reported results

O1. The ITP worked on formalizing the levels of interaction between the robot and the human and relied on the studies in the literature to provide a taxonomy for task and function allocation. A questionnaire was provided to the ITPs and to a group of students to understand the human involvement level in robotics applications.

(Completed: 100%)

O2. The ITP collected the results of their questionnaire that was completed by the ITPs and a group of students (as the number of responsive ITPs was not very high). After reviewing the results from this study, a clear need for guidelines for taking human factors into account during robotic system development can be identified. The material can be made accessible for the RobMoSys team as well. Although the need for more human factors engineering has become apparent, the study's significance suffers from the low return rate. It is possible to argue that there is need for more general information, whereas the low response rate makes it hard to directly identify specific methods, that are valid for a sufficiently large group. The result of this ITP show how human factors can be included into the development phase.

(Completed: 50%; Due to covid-19 restrictions)

Coach's perspective

The results of this ITP are important for RobMoSys. Taking into account the human factor at the design phases and modeling the operator skills offer several application possibilities like task allocation, task replanning, etc. The ITP involvement was delayed by one month because of a lack of resources in addition to the covid-19 pandemic restrictions. The workshops and visits planned by the ITP could not take place. The results are satisfying but additional work needs to be done for implementing the approach into the tools and applying the approach on specific use cases.

3.3.4 OPC UA for RobMoSys (OPC UA for RobMoSys)

ITP OPC UA for RobMoSys (OPC UA for RobMoSys) is conducted by Systerel.

The coach assigned to this ITP was Prof. Christian Schlegel (THUlm).

Project objectives

The objective of the project is to extend OPC UA capabilities of RobMoSys project. Indeed, OPC UA has been identified as a key vector for RobMoSys dissemination. Our OPC UA and cybersecurity expert will help RobMoSys members on the subject.

O1. Discuss with RobMoSys partners OPC UA requirements and set up a RobMoSys environment.

O2. Evaluate the existing OPC UA implementation (e.g., functionalities, security level).

O3. Suggest new features to members with associated deadline and cost. And implement new features after partners' agreement.

O4. Present the results in conferences and to the RobMoSys consortium.

Reported results

O1. This task was greatly rerouted to give more time to complete the objective O2, as no direct needs were identified within other RobMoSys partners (training, debug, ...).

(Completed: 100%)

O2. This task led to the production of a complete report detailing our findings about the OPC UA library that is currently used in the SeRoNet/SmartMDSD toolchain, and also about the wrappers that use it. This report is completed by a successful experiment using an off-the-shelf programmable logic controller that is connected to RobMoSys components through OPC UA.

(Completed: 100%)

O3. This task was used to test secure OPC UA connections with RobMoSys components. As agreed with our coach, this task is in the state of a "proof of concept", just to prove that secure connections work, without the hassle of modifying the SmartMDSD toolchain to add security parameters properly.

(Completed: 100%)

O4. This objective is greatly impounded by the COVID-19. However, the results were presented to the RobMoSys consortium within the virtual ITP workshop.

(Completed: 50%; Due to COVID-19 restrictions)

Coach's perspective

The ITP fulfilled the role of an advisor with respect to OPC UA in a very valuable way. Driven by technical insights resulting from the close collaboration between the ITP and the coach, the decisions about refinements of the proposed work had always been achieved in mutual respect of the respective expertise. Objective 1 had been rerouted in mutual agreement and with benefit for RobMoSys. Objective 4 got greatly impounded by COVID-19 which is to our regret but beyond our sphere of influence. The ITP provided very valuable insights, feedback and contributions to RobMoSys.

4 Conclusions

The final results of these ITPs are in general positive. Most of the ITP teams achieved their key objectives and uploaded their deliverables. The RobMoSys coaches have high expectations from the ITPs given their expertise, experience and initial interactions. The crucial role of individualized coaching provided by the experts of the RobMoSys core scientific members is an important contribution to the ITP success especially given the fact that following the RobMoSys approach is a paradigm shift and requires deep understanding of the underlying RobMoSys concepts.

Some issues have arisen during this period, with most of them related to the adaptation of the prepared solutions into the RobMoSys framework. Also, some projects experienced delays caused by unforeseen complexity of the goals presented in the original proposals. However, the involvement of the coaches and their individualized approach to the ITPs help finding solutions to these issues.

The RobMoSys community is getting stronger since the new ITPs are using the results of former RobMoSys ITPs to connect their work to the RobMoSys Ecosystem. This means the understanding of RobMoSys paradigm is improving among roboticists. However, it is necessary to increase the awareness and understanding within the wider audience to facilitate the future spread of the RobMoSys approach.