Deliverable: D1.4

Standardisation Activities Mid-Term Report

Deliverable Responsible: fortiss, DE
Version: 1.3

29/03/2017

Dissemination level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	X
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

Project Information

Acronym	openMOS
Name	Open dynamic manufacturing operating system for smart plug-and-produce automation components
Theme	FOF-11-2015: Flexible production systems based on integrated tools for rapid reconfiguration of machinery and robots
Grant agreement	680735
Start date	1, October 2015
Duration	36 months

Contact Information

Company Name	fortiss - An-Institut Technische Universität München
Address	Guerickestr. 25 80805 Munich Germany
E-Mail	info@fortiss.org
Phone	+49 3603522 0
Fax	+49 3603522 50

Version Control

Version	Date	Change
0.1	15/02/2017	Initial creation of the document structure
0.9	10/03/2017	First version
1.0	16/03/2017	Revised version merged with the feedback from Introsys
1.1	17/03/2017	Revised version merged with the feedback from Ford
1.2	22/03/2017	Revised version merged with the feedback from LiU
1.3	29/03/2017	Final version

List of Authors

Name	Role	Affiliation	Email
Chih-Hong Cheng	Author	fortiss, DE	cheng@fortiss.org
Kirill Dorofeev	Author	fortiss, DE	dorofeev@fortiss.org
Magno Guedes	Author	Introsys, PT	magno.guedes@introsys.eu
Stefan Profanter	Author	fortiss, DE	profanter@fortiss.org
Luis Ribeiro	Author	Linköping, SE	luis.ribeiro@liu.se
Alois Zoitl	Author	fortiss, DE	zoitl@fortiss.org
Tim Briggs	Contributor and Reviewer	Ford, UK	tbriggs5@ford.com

Table of Contents

1. Introduction

This document has been produced by partners in the openMOS consortium to report intermediate progress in terms of standardisation activities. While the ultimate aim is to successfully transfer results derived from openMOS to other emerging or changing standards, it is also very important to ensure that the proposed openMOS system architecture is compatible with existing or developing standards driven by other consortiums.

The following is an extract from the openMOS description of work that states the objectives of the task that includes the production of this deliverable:

> "**Task 1.3: Standardisation [fortiss, M3-M36]**
> Involved partners: Fortiss (lead), IntRoSys, Elrest, Ford, SenseAir, Inotec, Afag, HSSMI, UniNova, Lboro, KTH, LiU, Masmec, Asys, Electrolux, We+
>
> The main goal of this task is to increase industrial adoption and visibility of openMOS project results by incorporating existing and emerging industrial standards required in the envisaged application domains as well as actively providing input to standardisation organisations and disseminating standardisable project results as open standards. Hereby, our roadmap is to start with publicly available white paper standards which can then be transferred to the according standardisation bodies. This has been shown successfully by the AutomationML consortium."

Due to numerous standardisation activities at both national and EU level, it is impossible for the consortium to participate in all organizations. To this end, we have been selectively participating, with the goal of maximizing existing resources.

2. Activities conducted within the consortium

In the following sub-sections, we summarize standardisation activities we have been conducting or where we have contributed.

2.1. IEC 61499

Standard and one line summary
IEC 61499 Component solution for distributed industrial automation systems aiming at portability, reusability, interoperability, reconfiguration of distributed applications.
Level of participation (active / passive), and the underlying rationale
Since 2008 A. Zoitl from fortiss is an active Member of the IEC TC65B/WG15, which develops and maintains IEC 61499. As IEC 61499 is of great importance for fortiss not only in openMOS but also in other projects A. Zoitl took over the convenorship of this working group in 2015. This active role allows fortiss to bring issues and new topics identified in the projects into the standardisation process and actively develop the modelling language defined in IEC 61499.
Progress regarding bringing techniques from openMOS consortium to standardisation community
[Mapping skills to the modelling elements of IEC 61499] A key element of openMOS is to design mechatronic components and aggregations of mechatronic components (e.g., modules, machines) in a service oriented way where each of the elements offers its functionality in the form of skills. In openMOS we leveraged the event-based modelling language to implement the skill based interface as part of the control application and show how skill invocation and real-time control can be a part of the same control model. A key modelling element in this respect is IEC 61499's adapter interface. How to utilize this for skills and skill interfaces is shown in design patterns and examples as part of the open source project Eclipse 4diac. [Utilizing OPC UA in IEC 61499] In openMOS it was shown for the first time how the industrial communication protocol OPC UA can be used with IEC 61499, not only for the classical data exchange task but also for remote skill invocations. A reference implementation was done with the open source projects Eclipse 4diac and open62541, and also made available as part of the 4diac project to other IEC 61499 experts and users for review.

[Identifying Shortcomings] With the first experiments and demonstrators done in openMOS also shortcomings in the expressiveness of IEC 61499's modelling language were also identified. An example is that the state machines in IEC 61499 basic function blocks have no hierarchy. For certain application scenarios (e.g., common error handling code) this can lead to very large state machines. Such issues are brought to the attention of the working group and are considered in the new part of IEC 61499 – Part 5: Proposed Extensions – which is currently being progressed started by the IEC 61499 working group.

Synergy brought from the standardisation activities to the openMOS consortium

IEC 61499 is an important stepping stone for modelling the real-time control applications in openMOS, especially as the distributed and event-based nature of IEC 61499 matches extremely well to the skill based mechatronic component approach. As IEC 61499 is an industrial automation domain standard, it increases the acceptance of openMOS technology by industry.

Strategic outlook – How to further move openMOS technologies towards standardisation

Apart from identifying further shortcomings in IEC 61499's modelling language, further developments will focus on implementing the skill interface and the skill invocation/communication with the means of IEC 61499. This will include the following topics:

- Formally define how the adapter interface concept can be used for skill interfaces, not only in static interface description but also for a dynamic interface description (i.e., interface contract).
- Define how adapter interfaces can be mapped on the generic communication concepts defined in IEC 61499-1 Annex F and propose a new section for this annex as part of IEC 61499-5.
- Develop an IEC 61499 compliance profile for OPC UA describing how IEC 61499 can utilize the communication means of OPC UA for data exchange and the skill communication.

2.2. Automation Markup Language (AML)

Standard and one line summary

Automation Markup Language (AML) organization

AML is a neutral data format based on XML for the storage and exchange of plant engineering information.

Level of participation (active / passive), and the underlying rationale

Since December 2016, fortiss is now officially part of the AML consortium, with rights to participate in member meetings and to propose research agenda topics.

Due to our new member role, we are currently only moderately active in the consortium. When the openMOS mid-term demonstrator is available, we will be more active in bringing openMOS aspects to the consortium.

Progress regarding bringing techniques from openMOS consortium to standardisation community

[Reference converter implementation] As the AML standard has been established and the focus now is towards applicability and the connection to other standards within the openMOS scope, we have been focusing on open-source software contribution to the standardisation community.

We take the AML to OPC UA conversion companion, and create a reference converter implementation, where one can automatically convert the model description in AML to Java code for Eclipse Milo OPC UA Server Information model.

This implementation allows end-users to concentrate on developing models in AML format, and the code generation feature is automatically handled by our converter software.

Currently, we have compiled the source code and offered it as a binary available for download. We will consider how to further resolve issues related to licensing of open source software under the openMOS scope. At a minimum we plan to offer it as a free software to AML foundation members.

[Issues in the conversion process] We have also identified that in the AML to OPC UA conversion companion, there is no mapping from AML model to a member function in OPC UA. This creates some ambiguity in the conversion process. We have been raising this issue with the community.

Synergy brought from the standardisation activities to the openMOS consortium

AML has greatly influenced the way that the openMOS consortium creates the information model. We try to be compliant with AML, where AML has been the unified information exchange format for openMOS system being demonstrated.

Strategic outlook – How to further move openMOS technologies towards standardisation

Create a whitepaper to inform the AML foundation of the availability of the AML/OPC-UA converter software.
The offering of such an AML to OPC UA converter is considered as the unique contribution of fortiss to the AML consortium. There are two further action items in this aspect:

- In April 2017, we will release our prototype demonstrator used for mid-term review. As the software is automatically generated from the AML source code, the software demonstrator will be advertised in the AML consortium, to advocate the use of device adaptors following the openMOS methodology.

- We will also extend the tool to allow code-generation in C language such as the open62541 stack or the commercial stack from Prosys.

2.3. OPC Foundation and OPC UA

Standard and one line summary

OPC UA

OPC Unified Architecture (OPC UA) is a machine to machine communication protocol for industrial automation

Level of participation (active / passive), and the underlying rationale

Within the openMOS consortium, fortiss has recently joined the OPC foundation. We have been relatively active in open source community activities related to OPC UA, but due to our new member role, we are still relatively inactive overall.

Progress regarding bringing techniques from openMOS consortium to standardisation community

openMOS consortium members have been very active in contributing to the open source development of OPC UA implementation. This is to fulfil the "openness" perspective for openMOS, where the developed ingredients and the integrated technologies can be reused by the community when possible. In the following, we summarizes our activities.

- Contribution to open62541- open source C implementation

  - open62541 did not support OPC UA Local Discovery Services (LDS) as defined in Part 12 of the OPC UA Specification. Our main contribution includes the implementation of the LDS into open62541. After basic evaluation of different mDNS implementations for use in the open62541 project, the code base was extended to provide the required services and data structures. This also included extending various components of the stack to ensure its compatibility with the OPC UA specification. Numerous bug-fixes have been submitted and code refactoring/improvements are already integrated in the next release. The LDS implementation has also already been merged into the master branch (https://github.com/open62541/open62541/pull/732, 3000 lines of code addition) and will be included in the next major release 0.3. Testing the implementation in combination with other different implementations and in the internally organized workshop showed its robustness and advantages for openMOS.

- Contribution to Milo - open source Java implementation as supported by the Eclipse Foundation
  - Milo did not support OPC UA Local Discovery Services (LDS) as defined in Part 12 of the OPC UA Specification. To enable auto device discovery for openMOS we implemented these services according to the Specification and added them as a Pull Request to the original Milo source code (https://github.com/eclipse/milo/pull/89, 1800 lines of code addition). During extensive testing we also fixed some bugs in the base Milo implementation and reported these fixes back to the original project, where they are now integrated into its main branch.

Synergy brought from the standardisation activities to the openMOS consortium

At the start of openMOS project, there were technical debates over technologies to be applied in the manufacturing service bus (MSB). By understanding the benefits that OPC UA can bring, we have been advocating the use of OPC UA technologies and bring it to the openMOS context.

As an example, towards the mid-term review, we have been building a demonstrator solely using the OPC UA technology. This implies that both the manufacturing service bus and the device adaptor (to connect generic machine to the MSB under openMOS context) are implemented using the technology.

Strategic outlook – How to further move openMOS technologies towards standardisation

Currently in OPC UA standardisation, there are three ongoing activities in which we plan to actively participate

- The OPC foundation is now pushing the *publish-subscribe* mechanism. We will join the development of such standard revision activities, as having capabilities to perform publish-subscribe is important from the openMOS perspective.
- Another aspect is regarding the *real-time capability* of OPC UA. Currently, there is an ongoing development to investigate how time-sensitive networking (TSN) can be used jointly with OPC UA, in order to make statements over the quality of service such as guaranteed response time. Within the consortium, fortiss is developing such a demonstrator (out of the openMOS project scope). We will also actively participate in activities when possible.
- Apart from pushing items in the standard, we expect to influence the members by showing our demonstrators and openMOS design documents as "recommended practices".

Lastly, we will continue contributing to the open source development of the OPC UA software stacks such as Eclipse Milo and open62541.

2.4. IEC TC65 AHG 3

Standard and one line summary

IEC TC65 Ad hoc group 3

Smart Manufacturing Framework and System Architecture

Level of participation (active / passive), and the underlying rationale

Moderate participation. Currently the key activity is only about collecting use cases and existing standards. The meetings and the teleconferences are attended, but the standardisation activity has not yet reached the stage of designing architecture documentation.

Progress regarding bringing techniques from openMOS consortium to standardisation

As currently the activity only involves collecting standards and use cases, until we have finished the mid-term demonstrator, the consortium does not plan to play an active role.

Synergy brought from the standardisation activities to the openMOS consortium
None so far.
Strategic outlook – How to further move openMOS technologies towards standardisation
The openMOS consortium will contribute to the use case when the mid-term demonstrator is constructed. Apart from contributing to the use case, the standardisation group will soon start the activity of creating a reference architecture, including an intelligent manufacturing information model. When the process starts, we will contribute our existing architecture documentation, hoping it will serve as the basis architecture to be iterated.

2.5. IEEE P2660.1

Standard and one line summary
IEEE P2660.1 - Recommended Practices on Industrial Agents: Integration of Software Agents and Low Level Automation Functions This recommended practice aims to solve the interface problem when integrating industrial agents with automation equipment in the context of cyber-physical systems. It will provide a collection of best practices for this integration process aiming to standardise the interface itself in order to allow its reuse and improve transparency.
Level of participation (active / passive), and the underlying rationale
The IEEE P2660.1 activity is focusing on the analysis of practices that have already been documented and have reached at least a proof of concept maturity stage. The developments within openMOS are, in general, aligned with the practices currently under analysis but since they are under development these activities have not been submitted to the IEEE P2660.1 for consideration in the analysis.
Progress regarding bringing techniques from openMOS consortium to standardisation
To be most likely considered in later stages of the project.

Synergy brought from the standardisation activities to the openMOS consortium
Current developments are, in general, aligned with the IEEE P2660.1 activities.
Strategic outlook – How to further move openMOS technologies towards standardisation
The technology being developed under the openMOS project, once validated and documented, may be incorporated in the results of the IEEE P2660.1 activity.

2.6. ROS

Standard and one line summary
ROS Robot Operating System is a software framework that joins a vast set of libraries and tools for developing robot applications. It is widely used in the mobile robotic community and is being gradually introduced in the industrial manufacturing industry.
Level of participation (active / passive), and the underlying rationale
The openMOS consortium is creating an AGV for proving Plug-and-Produce capability of mobile transportation systems. This AGV is fully compatible, at the low level with the ROS framework. The consortium is developing, in parallel, a module for enabling the ROS standard services to understand the skill-oriented paradigm adopted by openMOS devices and report status feedback to the upper openMOS layers.
Progress regarding bringing techniques from openMOS consortium to standardisation
Due to the quick adoption and potential of the ROS technology regarding robot control, it is seen by the consortium a clear advantage to enable such technology to interact with the openMOS system. By making these interfacing tools available, it is expected that as well as it being appealing for the current users of ROS enabled devices to adopt the openMOS technology, but also that system integrators see the synergies between the ROS and openMOS technologies as a competitive solution for implementing new production lines.

Synergy brought from the standardisation activities to the openMOS consortium

The adoption of ROS has enabled, so far, the quick development of a full control stack for a newly created robotic device (the AGV). The respective tools, such as its simulation environment, enabled the preliminary test and validation, not only of the mentioned control stack, but also of the openMOS technology that interacts with the AGV.

Strategic outlook – How to further move openMOS technologies towards standardisation

It is expected that any ROS enabled robot will be able to expose "skills" and execute commands ("recipes") to/from the openMOS system. For this, only one requirement must be met:

- The ROS enable robot should make use of the standard way proposed by the ROS practices for interacting with external modules that is, using typical nomenclature for topics (e.g. \cmd_vel, \odom, etc.) and standard message types.

After testing the ROS-to-openMOS interfacing module with the openMOS AGV, several tests will be carried out using different robot stacks, available online in the ROS webpage, in a simulated environment, to prove the ability of the system to integrate other third-party ROS enabled platforms.

3. Concluding Remarks

Overall, the openMOS consortium has been actively contributing to standards that are highly relevant to the proposed technology stack such as OPC UA and AML. The system design as well as the demonstrator are largely adapting these standards.

One of the important pieces of work to be conducted in the later phase of the project execution is to introduce our reference architecture and implementation to the standardisation organization, which is only possible after we have completed our mid-term prototype demonstrator. The feasibility of the proposed architecture can only be truly confirmed once implementation has begun. Our mid-term demonstrator will be advertised as a recommended practice of how the developed standards can be applied.