RobMoSys

H2020—ICT—732410

RobMoSys

Composable Models and Software for Robotics Systems

Deliverable D7.8
Business Models for the Ecosystem – M48

Huáscar Espinoza (CEA), Marco Jahn (EFE), Anna Principato (TUM)

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No. 732410

Project acronym: RobMoSys

Project full title: Composable Models and Software for Robotics Systems

Work Package: WP 7 - Exploitation

Document number: D7.8

Document title: Business Models for the Ecosystem – M48

Version: 1.0

Due date: December 31th, 2020

Delivery date: 31.12.2020

Nature: Report (R)

Dissemination level: Public (PU)

Editor: Huáscar Espinoza (CEA)

Author(s): Huáscar Espinoza (CEA), Marco Jahn (EFE), Anna Principato (TUM)

Reviewer: Herman Bruyninckx (KU Leuven)

Executive Summary

This document addresses business-related issues of the RobMoSys ecosystem, particularly focused on an open source and open assets strategy. The proposed business models have been incrementally specified and continuously aligned to market needs during the RobMoSys project. This deliverable represents the final version of **Business Models for the RobMoSys Ecosystem (Deliverable D7.8)**, as part of work package **"Exploitation" (WP7)**.

The main topics of this document are:

1. Background on business models, open source business models and licenses types.
2. Market analysis providing a systematic specification of the needs and value chains of the different types of RobMoSys stakeholders, including tool vendors, integrators, system and OEM providers, certification entities, and standardization bodies.
3. Specifying a final version of the Business Model fo RobMoSys.

This first version of the Business Models for the Ecosystem (Deliverable D7.2) was focused on establishing the basics and on specifying the available knowledge of markets, stakeholder needs and best business approaches for the RobMoSys partners. This scope has been extended in the second deliverable (Deliverable D7.5) by adding the feedback from Tier-1 experts coming from the business world and taking into account the return of experience from related projects. We also considered an internal workshop to define the value proposition of RobMoSys. The final version of the RobMoSys Business Models (current Deliverable D7.8) provides a view of the RobMoSys strategy for the open source community from the business point of view.

RobMoSys looks for the 'snowball' effect: the project attracted early adopters with the quality of code, the initial assets attached to the code like the Wiki guide, documentation, scientific and technical papers, first releases, well defined code infrastructure, and the interest of the adopters will do the rest.

Content

1 Introduction

RobMoSys is about managing the interfaces between different roles (robotics expert, domain expert, component supplier, system builder, installation, deployment and operation) and separating concerns in an efficient and systematic way by making the step change to a set of fully model-driven methods and tools for composition-oriented engineering of robotics systems.

RobMoSys's vision is that of an agile, multi-domain, model-driven European robotics software ecosystem. It will consist of specialised set of players with both vertical and horizontal integration 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.

Popular **digital platforms in robotics** (such as ROS and i-Cub) pride themselves to unite communities of hundreds of stakeholders, which need to be preserved and strengthened. However, these communities are still rather fragmented, by representing specialized customer groups interested by a specific technology or domain. To strengthen the established platforms, to enable interconnections between them, and definitely also between new ones in other domains (not in the least, the application domains of big potential new end-users), RobMoSys envisions an integration approach built on-top-of, or rather "around", the current code-centric platforms, by means of the systematic development and application of **model-driven methods and tools that explicitly focus on system-of-system integration, at all levels of abstraction and interaction**, hence not just software code.

From a **business perspective**, changes are expected in the evolution of traditionally linear supply chains into complex, dynamic, and connected value webs (co-creation and collaboration). New models of cooperation that depend on seamless integration of diverse partners require new ways of early involvement of customers and business partners into design- and value-adding processes. Digital platforms represent a key enabler to facilitate greater levels of connectivity, collaboration and co-creation with other businesses.

RobMoSys aims at a disruptive change in the approach to robotic software related business by establishing a **common methodology** based on the use of composable software models and by nourishing an **ecosystem** of methodology-based tool chains to support the implementation of the methodology. To cover the largest possible spectrum of stakeholder groups, we envisage to keep a high degree of **openness**, both of tools and also of assets (e.g. models, patterns and libraries).

This document addresses business-related issues of the RobMoSys ecosystem, particularly focused on an open source and open assets strategy. The proposed business models have been incrementally specified and continuously aligned to market needs during the RobMoSys project. This deliverable represents the final version of **Business Models for the RobMoSys Ecosystem (Deliverable D7.8)**, as part of work package **"Exploitation" (WP7)**.

1.1 Scope

This first version of the Business Models for the Ecosystem (Deliverable D7.2) was focused on establishing the basics and on specifying the available knowledge of markets, stakeholder needs and best business approaches for the RobMoSys partners. This scope has been extended in Deliverable D7.5, by adding the feedback from Tier-1 experts coming from the business world and taking into account the return of experience from related projects. The final version of the RobMoSys Business Models (current Deliverable D7.8) provides a view of the RobMoSys strategy for the open source community from the business point of view.

1.2 Document Structure

The remaining of this document is organised as follows:

- Section 2 provides a background on business models, open source business models and licenses types.
- Section 3 presents a market analysis providing a specification of the needs of the different types of RobMoSys stakeholders, including tool vendors, integrators, system and OEM providers, certification entities, and standardization bodies.
- Section 4 addresses the business model proposed for RobMoSys focused on the open source community creation and aligned to the market needs.

2 Background

2.1 What is a Robotics Software Platform?

By a “robotics software platform” we mean a software package which simplifies programming of several kinds of robotic devices by providing:

- a unified programming environment;
- a unified service execution environment;
- a set of reusable components;
- a debugging/simulation environment;
- a package of “drivers” for most wide-spread robotics hardware
- a package of common facilities such as computer vision, navigation or robotic arm control

The cost of control software accounts for a large share of the overall cost of a typical robotics project. For example, up to 80% of an industrial automation project is spent on system integration which includes software development/customization. So, the main idea behind any robotics software platform is to simplify the job of robotics software engineers – and thus reduce the project costs.

2.2 Open Source Concepts

2.2.1 Elinor Ostrom Principles

Elinor Ostrom, Nobel price of Economy in 2009, designed eight principles for managing stable Common Pool Resource (CPR):

1. **Clearly defined** (clear definition of the contents of the common pool resource and effective exclusion of external un-entitled parties);
2. The appropriation and provision of common resources that are **adapted to local conditions**;
3. **Collective-choice** arrangements that allow most resource appropriators to participate in the decision-making process;
4. **Effective monitoring** by monitors who are part of or accountable to the appropriators;
5. A scale of **graduated sanctions** for resource appropriators who violate community rules;
6. Mechanisms of **conflict resolution** that are cheap and of easy access;
7. **Self-determination** of the community recognized by higher-level authorities; and
8. In the case of larger common-pool resources, organization in the form of **multiple layers of nested enterprises**, with small local CPRs at the base level.

Deciding to contribute to the open source either as a consumer of open source components or producer of open source components or both is a volunteer act to manage **in common** some software code and all the resources attached to this software.

Because the initial investment might feel harder, it is important to feel guided by Elinor Ostrom principles.

2.2.2 What is Open Source Software?

The Open Source Initiative (https://opensource.org) provides a very good definition of Open Source Software (OSS) and defines it in 10 commandments (https://opensource.org/osd-annotated):

1. **Free redistribution**: The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
2. **Include source code**: The program must include source code, and must allow distribution in source code as well as compiled form.
3. **Modifications and derived works**: The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4. **Integrity of author’s source code**: The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time.
5. **No discrimination against person and groups**: The license must not discriminate against any person or group of persons.
6. **No discrimination against fields of endeavor:** The license must not restrict anyone from making use of the program in a specific field of endeavor.

7. **Distribution of license**: The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. **License not specific to a product**: The rights attached to the program must not depend on the program's being part of a particular software distribution.
9. **License not restricting other software**: The license must not place restrictions on other software that is distributed along with the licensed software.
10. **License technology neutral**: No provision of the license may be predicated on any individual technology or style of interface.

2.2.3 Why Do we Need Open Source?

From Small and Medium-sized Enterprises to large organizations, a lot of companies have adopted and contribute to the one or more open source communities like the Apache Software Foundation, the Eclipse Foundation, and the Linux Foundation. Some of them have been involved for several decades already. This choice has nothing to do with altruism: it is a business strategy.

Actually, organizations like open source software to proprietary software for many reasons, including:

- **Maturity of the model**: There are numerous examples of projects and products based on the OSS which are more reliable and sustainable than other proprietary solutions. Here are a few of them:
  - The Linux operating system, which is now widely adopted by major private or public organizations.
  - Apache HTTP Server, certainly the most used HTTP server. It played a key role in the initial growth of the World Wide Web.
  - The Mozilla web browser called Firefox, which in 2009 was the most popular web browser with 32% of the market. In 2016, between 9% and 16% of individuals use Firefox as their desktop browser, making it the second most popular web browser, the first one is Google Chrome.
- **Cost of acquisition**: Adopters of OSS obtain a financial gain for each stage of a project. E.g.:
  - **Free**: it’s free to download and use.
  - **Try before buy**: because it is free, companies can try different OSS solutions before making the decision to invest time or resources in a specific one.
  - **Hiring is easier**: because OSS is free, many developers use it and become proficient with the software early on in their career or during their studies. This makes it easier and less expensive to find good developers that have experience with the open source technologies they have adopted for their project.
  - **Training**: it’s easier to train a team with the assets produced by OSS community of developers.
  - **Customizability**: open source software can be tweaked to suit various needs. Since the code is open, it's simply a matter of modifying it to add the functionality needed by the project.
  - **Time to Market is shorter**: products don’t have to be built from scratch. Companies can rely on sustainable OSS and build their solution on top of it.
  - **Lower total cost of ownership**: companies can rely on OSS community for maintenance and, by joining the community, they mutualized maintenance costs.
- **Dependence**: Organizations don’t depend on the status of the subcontractor who originally built the software. In open source software, if a contributor ceases working on a project for any particular reason, the source code stays accessible and someone else can take over the work.
- **Quality of the code**: OSS gets closer to what users want because those users can have a hand in improving it.
- **Security**: Some users consider OSS more secure and stable than proprietary software, mainly because they can control the source code and they can identify and fix errors or omissions. The efforts are mutualized with the other community members, which results in secure and stable source code.

2.2.4 The License Spectrum

Regarding the licensing part, there is a broad spectrum of licenses, from permissive licenses (such as the MIT, BSD Style and Apache) to proprietary licenses, which typically do not allow users to modify or distribute the software.

In between, there are the “Copyleft Licenses” which offer the right to freely distribute copies and modified versions of software with the stipulation that the same rights must be preserved in derivative works created later. In contrast with permissive licenses, these are considered protective or reciprocal as they impose more constraints on the users or integrators of the software. Within this share of the spectrum we find both weak (e.g. EPL, MPL) and strong (GPL, AGPL) copyleft licenses.

The strength of the copyleft governing a work is an expression of the extent to which the copyleft provisions can be efficiently imposed on all kinds of derived works. Weak copyleft[5] licenses require users to redistribute their changes to the software under the same license, but are considered more business friendly as they typically allow the use of a different license (including sometimes proprietary licenses) when embedding the software in a larger product or solution (see Figure 1).

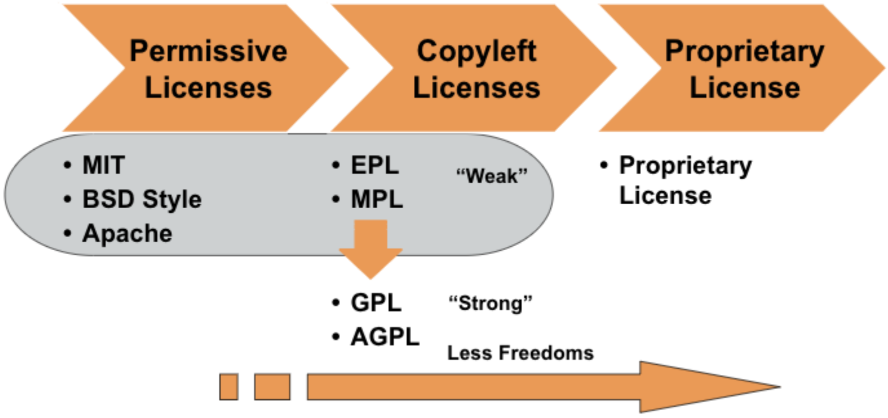

Figure 1: The License Spectrum

2.3 Towards Business-Friendly Open Source Ecosystems

Many observations show that it is possible to develop a business model with software licensed under any kind of open source licenses. RedHat is a good example of a company that generates a significant revenue from an offering built around software mainly licensed under Copyleft licenses. But it should be noticed that in the recent years, more and more open source projects use permissive or semi-permissive license as they are considered more "business-friendly". This is particularly the case for "platform" projects that are subject to be extended to create "products". A typical pattern is that the platform is made available under a permissive license (for example Apache or BSD), or a semi-permissive license like the Eclipse Public License, and some contributors create proprietary extensions that are marketed under a classical software vendor scheme.

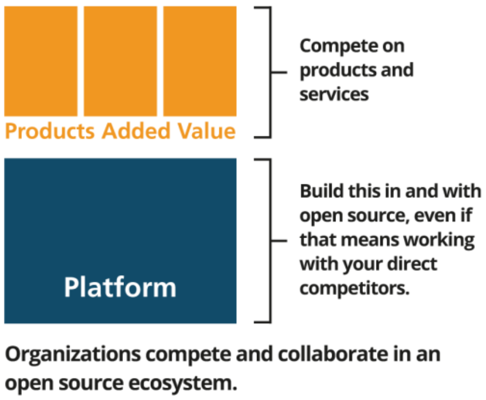

Figure 2: A business-friendly ecosystem based on extensible platforms

Figure 2 shows this scheme as it has been applied with success in the Eclipse ecosystem since 2004. This resulted in thousands of products built on top of the Eclipse platform both by Software vendors and by Software integrators.

2.4 Basics of Business Models

A business model can be described as the system of actions carried out by a corporation or community in the course of its economic activities. It can be decomposed in a number of building blocks. We describe the Business Model Canvas, which we will use for RobMoSys business models.

2.4.1 Business Model Canvas

The business model canvas [Osterwalder 2010] can summarize all the business-related concept descriptions in the right business context. In addition to the BOAT framework, the Canvas includes the value proposition, cost structure, and revenue streams. Together with the BOAT framework and e3 value models, the canvas covers all business aspects and the details to analyse the business case. The Canvas summarizes these results in one overview.

The Business Model Canvas is a strategic management template for developing new or documenting existing business models. It is a visual chart with elements describing a firm's value proposition, infrastructure, customers, and finances. It assists firms in aligning their activities by illustrating potential trade-offs. An overview of all these aspects is depicted in Figure 3.

The building blocks of the Business Model Canvas consist of:

1. **Customer segments**. Customers are the heart of the organisation. This building block defines the different groups of people or organisations that the business wants to reach with the product they offer. It is relevant to define different groups if the offered value needs to be a separate one, either in content, (consumption) channel, relationships, profitability, or different groups are willing to pay for certain aspects of the objects.
2. **Value proposition**. The building block value proposition describes the bundle of products/objects and services that creates value for the customer segments. It is the reason that customers prefer one business over another. The value proposition provides value through various elements such as newness, performance, customization, getting the job done, design, brand/status, price, cost reduction, risk reduction, accessibility, and convenience/usability.
3. **Channels**. The building block ‘channels’ describes how the business is communicating with its customers. These channels have different functions, like creating awareness about the products/objects offered, determining the value proposition in negotiations, buying products, delivering products, provide value proposition to the customer, as well as customer support.
4. **Customer relationships**. In order to optimize operations and reduce risks of a business model, organizations usually cultivate buyer-supplier relationships so they can focus on their core activity. Complementary business alliances also can be considered through joint ventures, strategic alliances between competitors or non-competitors.
   To ensure the survival and success of any business, companies must identify the type of relationship they want to create with their customer segments. Various forms of customer relationships include:
   - *Personal Assistance*: Assistance in a form of employee-customer interaction. Such assistance is performed either during sales, after sales, and/or both.
   - *Dedicated Personal Assistance*: The most intimate and hands on personal assistance where a sales representative is assigned to handle all the needs and questions of a special set of clients.
   - *Self Service*: The type of relationship that translates from the indirect interaction between the company and the clients. Here, an organization provides the tools needed for the customers to serve themselves easily and effectively.
   - *Automated Services*: A system similar to self-service but more personalized as it has the ability to identify individual customers and his/her preferences. An example of this would be Amazon.com making book suggestion based on the characteristics of the previous book purchased.
   - *Communities*: Creating a community allows for a direct interaction among different clients and the company. The community platform produces a scenario where knowledge can be shared and problems are solved between different clients.
   - *Co-creation*: A personal relationship is created through the customer's direct input in the final outcome of the company's products/services.
5. **Revenue streams**. If the customers are the heart of the business, revenue streams are the arteries of the business. An important question for the revenue stream is: what is the willingness-to-pay of each customer segment for the products offered. There are two kinds of revenue

streams: transaction based and revolving funds. A finer division is:

- *Asset Sale* - (the most common type) Selling ownership rights to a physical good. i.e. Wal-Mart
- *Usage Fee* - Money generated from the use of a particular service i.e. UPS
- *Subscription Fees* - Revenue generated by selling a continuous service. i.e. Netflix
- *Lending/Leasing/Renting* - Giving exclusive right to an asset for a particular period of time. i.e. Leasing a Car
- *Licensing* - Revenue generated from charging for the use of a protected intellectual property.
- *Brokerage Fees* - Revenue generated from an intermediate service between 2 parties, i.e. broker selling a house for commission
- *Advertising* - Revenue generated from charging fees for product advertising.

6. **Key resources**. The building block 'key resources' describes resources that will be required to create value for the customer. They are considered an asset to a company, which are needed in order to sustain and support the business. These resources could be human, financial, physical and intellectual.
7. **Key activities**. The most important activities in executing a company's value proposition.
8. **Key partners**. The key partners are the partners that will provide the knowledge, basic functionality, social networks, for the platform to run smoothly. Partnerships can be formed using strategic alliences, co-optation (a co-operation between competitors), joint ventures, and direct buyer-supplier relationships to secure product deliveries.
9. **Cost structure**. This describes the most important monetary consequences while operating under different business models.

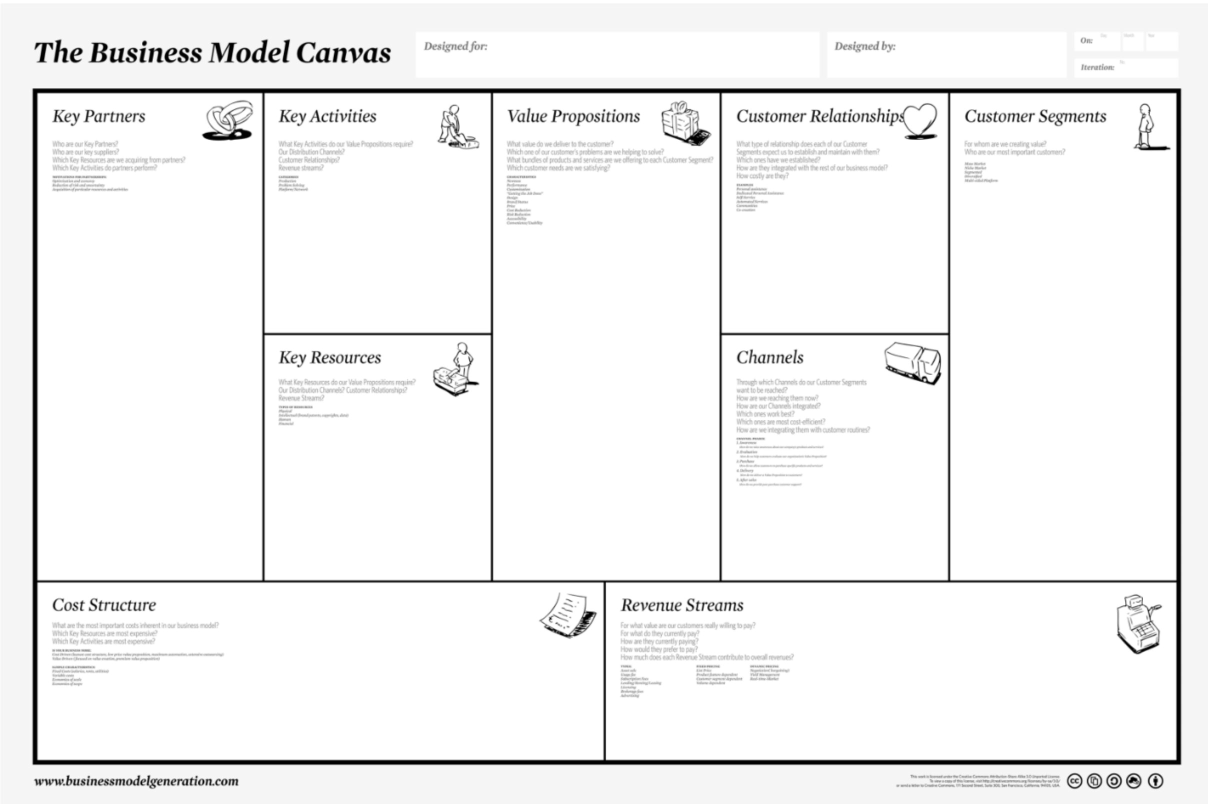

Figure 3: The Business Model Canvas

3 Market Analysis related to RobMoSys

3.1 Market Size and Share

3.1.1 Market size in robotic-related domains

Robotics is an enabling technology both for the manufacturing industry as well as a growing number of service sectors. Overall, a robotics-related increase of the EU GDP by EUR 80 billion by 2020 is expected by the robotics community in their strategy research agenda [SPARC 2013].

Robotics technology can be deployed in a wide range of different market domains. Each domain has its own needs and requirements. The Multi-Annual Roadmap for Robotics 2020 [Roadmap 2015] provides an overview on the technology trends and market opportunities for the most important robotics domains.

The by far largest and most important application domain today is industrial robotics – robots deployed in the manufacturing environment. With $13.1B of revenues in 2016, industrial robots contributed to 61% of the global overall robotics market [IFR-WorldRobotics 2017]. Largest industries applying industrial robots are the automotive and electronics industry. Main driver for global growth is the electronics industry.

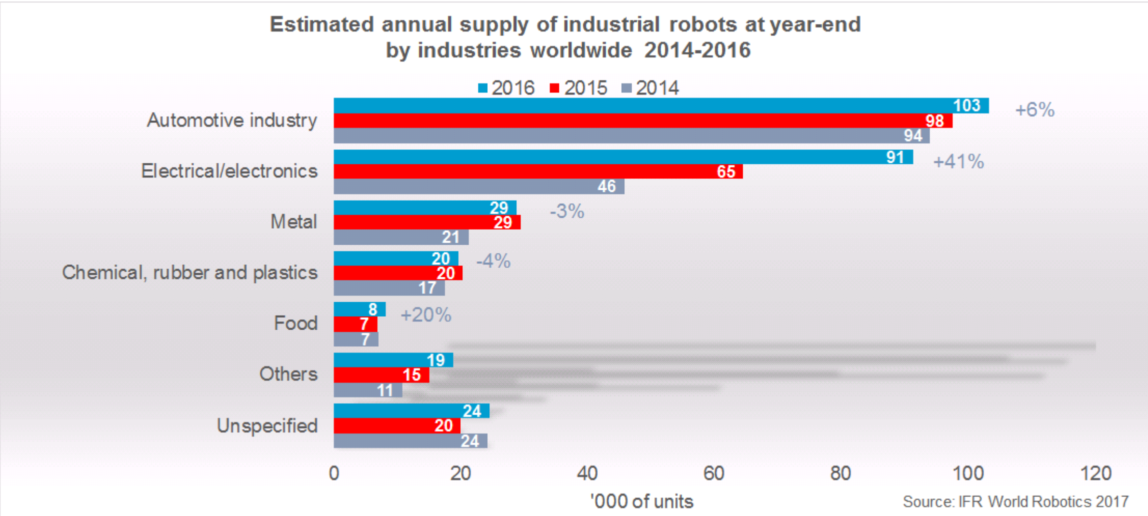

Figure 4: Main application industries of industrial robots [IFR-WorldRobotics 2017]

Professional service robotics make up about 22% of today's global robotics revenues. Main application areas are medical, logistics, field and defence. The costs per system vary drastically for the different application areas -from ~$16,000 on average for a public relations robot to $1M for a medical system.

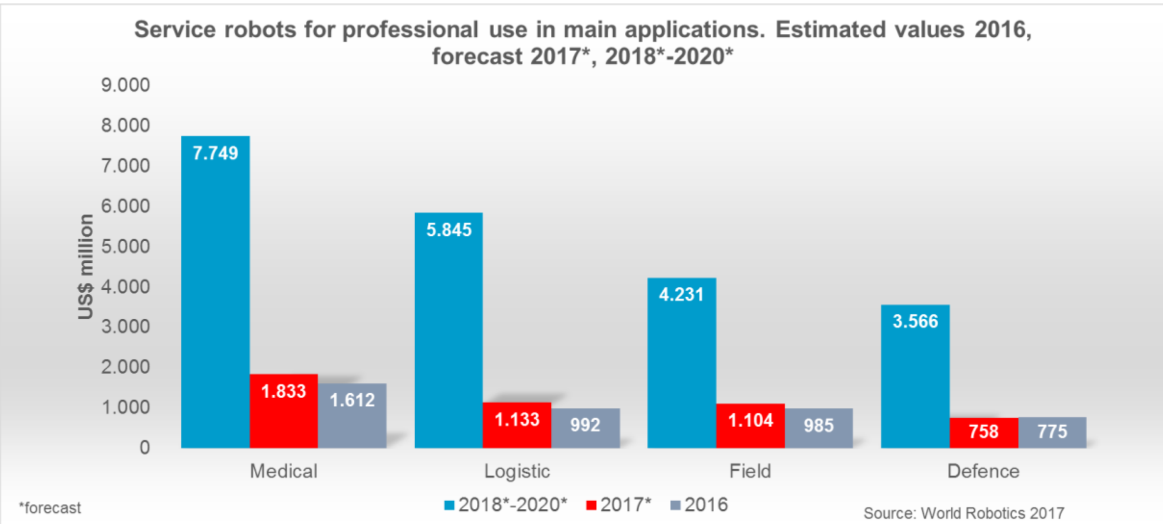

Figure 5: Main application sectors for professional service robotics [IFR, WorldRobotics 2017]

Robots for personal and domestic applications and entertainment make up the third group. Unit prices here are even less, they already have become commodities.

The robotics market is complex involving a diverse range of opportunities. Organisations may create value by concentrating on specific end applications, supplying different types of robot, modules, sub-systems, tools, or providing services within the market. It also includes dedicated supply chains, design services, and research and development organisations. Providing a coherent categorisation of the potential in each type of market is an important step in evaluating the potential for robotics and robotics technology.

3.1.2 Market size in robotic software and services

The robotics hardware market will grow from $21.4B (according to the IFR, the International Federation of Robotics) in 2016 to $40.1B in 2020, growing at an average compound annual growth rate of roughly 15% [Murphy 2017]. This estimate only includes hardware sales, not software or other supporting services for robotics. When factoring in these additional markets, it is estimated that the total robotics market value could be 3x larger. It is anticipated that hardware and components used for robotics will largely be commoditized over the next 10 years. Differentiated hardware will be the exception to the rule and unique value will be driven by software and accompanying services.

Not all robots are deployed once and then used over their complete lifetime. This leads to the development of new business models, away from classical sales to lease or rent models ("Robots as a service", RaaS) or even the complete service including operators is rented (e.g. for maintenance and inspection tasks).

The report on robotic software platforms market [Technavio 2016] estimates that this will reach USD 8.15 billion by 2020, should the market sustain a CAGR of 4.57% during the forecast period.

3.1.3 Market size on software and modelling tools

Analysts' reports, as the VDC report [Girard 2016], project that the global software and system modelling tool (SSMT) market will expand from $916.3M in 2014 to reach $1228.8M in 2018, a Compound Annual Growth Rate (CAGR) of 7.6%. Increasing system complexity is the primary driver of modelling tool revenue growth in both the embedded and enterprise/IT market segments. The volume of new users is expanding as any organizations find modelling tools are an effective approach for overcoming the challenges their software developers or engineers face designing increasingly complex software and systems. Extensive process standards and regulatory requirements will continue to be leading drivers of modelling tool use in the embedded market, especially in safety-critical industries. The influence of mandates on development tool use is also rising in the enterprise market as financial reporting requirements become more common.

3.1.4 Market size on robotic software platforms

The demand for robotic software platforms is scaling up as they enable industries to reprogram the machines frequently, modify, and test their robots to suit multiple application demands. In 2015, the APAC region dominated the global robotic software platforms market with a 66% stake in the overall market share. The market is booming in this region as many manufacturers are adopting robots in their plants to increase productivity and meet the demands put forth by industries such as automotive and healthcare. Government orders and offers that encourage local robotics industry to boost exports, loans at low interest and other incentives for robotic factories, and rising labor wages are some of the factors that will contribute to growth in the market in this region in the coming years.

3.2 Market Trends

There are a number of market trends that confirm the market opportunities for the RobMoSys tool-chain(s), which will be valuable not only in robotics but in all markets where embedded systems are willing to incorporate autonomous capabilities.

3.2.1 Time to market & competitiveness

In an increasingly competitive market, with an ever-increasing speed of innovation, organisations must pursue easier, faster, and more efficient ways to build and assure robotic software systems to remain competitive. The cost of adopting a robotics system varies depending on the application and software account. Software platforms, such as RobotStudio offered by ABB or iRobot Aware by iRobot, can be used to program or test robotic automation systems. Compared to the traditional teach pendant method that consume 4-40 hours, these software platforms allow offline programming, and therefore are more feasible and reduce production downtimes. The offline program is coded on a personal computer using C, C++, and Python or URBI, depending on the vendor's preference.

3.2.2 Reuse and cross-domain harmonization

The trend and need is to increase reuse-geared development and integration processes as a major means to reduce costs. The challenge with reuse is not necessarily with the reuse itself but with reuse of composable artefacts, especially where artefacts are cross-domain. A robotics software artefact for manufacturing robots might not be able to be applied directly “as is” in the healthcare or agriculture domains for example.

The most important goal of any robotics platform company is to see their product being pre-installed on a mass-produced commercial robot (just like Windows is pre-installed on new PCs). This is still yet to happen.

The RobMoSys approach will also help robotics technologies to be accepted in safety-critical domains (automotive, railways, etc.) and applications, and would prepare robot technologies to certification. Robotic software, by nature, inherits the requirements of the system in which it should work. When “immersed” in a specific domain, the development practices of the hosting domain will be immediately imposed to the robotic one. After some periods of disorientation arguing that what is done in safety critical domain to validate software cannot be done to validate robotic software, standardisation bodies will eventually put new requirements on methods for robotics software validation.

The risk that robotic technologies will remain only in particular niche markets, e.g. autonomous shuttles for private sites, or in established ones, such as industrial automation but without concrete perspectives of taking advantage of their real innovation potential (e.g. autonomous capabilities), is directly related to the capacity of robotics to embrace mature software and validation methodologies, as proposed by RobMoSys. Here, the availability of more formal models is expected to be the key differentiator.

3.2.3 Open solutions

The use of industrial robots is common in the automotive, metal, electronic, and healthcare industries. This rise is attributed to the technological advances and user friendly and simple programming software provided by third-party vendors and OEMs. The software platform and the training sessions, required before installing robots in industries, are expensive. However, the ease of offline programming, reduction in production and equipment downtime, and the profit earned have expanded the use of robots in these industries. The availability of open source software from third party vendors (such as OROCOS and OpenJAUS) that are compatible with robots of different makes are motivating end-users to use machines in complex manufacturing processes.

Increased adoption of software modelling tools, however, has not translated into equivalent increases in commercial market revenue to the degree that VDC report [Girard 2016] had previously anticipated. VDC suspects that a number of factors, including the availability of low-priced and open-source modelling tools, have significantly impacted the growth of the commercial market. Recent business trends are also encouraging proprietary language-based modelling tools move to common open-source platforms backed by private funding to allow tools to rapidly adapt to customer needs [Eclipse 2016]. RobMoSys objectives are completely in line with these trends.

On the industrial robotics side, most established manufacturers of industrial robots provide their own platforms for programming their robots (e.g. look at ABB’s robotics software products). They would probably prefer selling their own products other than using third-party products. To counter this tendency, OMG robotics group has recently staged an orchestrated campaign to create a set of standards which would break such vendor lock-in. The efforts are in the very early phase, but we hope they will produce working standards in medium-to-short term. A similar but independent development is going on in Europe – look at European Robotics Platform web site.

The revenue model of the software market is diverse and vendors compete on the basis of suitability for their business. The most preferred revenue model of the robot manufacturers is the hybrid model, wherein end-users are charged a fixed amount for software licensing, support and services, and management. But the easy availability of open source software from vendors such as OROCOS and Gazebo is encouraging end-users to reprogram and simulate robots in the plant. An open source software provider only charges for support and services opted by end-users

3.2.4 Increasing service life with shortening product lifecycles

The average service life of an industrial robot is estimated between 12 to 15 years [IFR, WolrdRobotics 2017]. That does not imply that robots are used in the same application, e.g. manufacturing process over the whole time. Considering the shorter product lifecycles, we see, for example in the electronics industry, that setups need to be reconfigured several times during the service life of the robot.

Given the rate at which technology is developing, tools, methods and documentation that are used for the original product need to be available throughout the product lifespan. This is so that incremental development and bug-fixes, for example, can be performed cost effectively. Moreover, the ease of re-usability and re-programmability becomes increasingly important to reduce the costs for resetting the production cell or service robot application.

3.2.5 Modularity

Advances in technology have transformed robotics from handheld/static instruments to remote technology-enabled machines and from teaching to self-learning entities. One such innovation is the self-configurable modular robot, which is created by connecting a number of modules with memory units and CPUs. These machines can change shapes depending on the task at hand. Each module of such new generation robots can communicate with the others and take decisions before performing a task.

Two of the best concepts in the robotics field that have emerged recently are modular and transformer robots. The transformer robot uses the same structure to make various forms. Such machines can change or transform to two different shapes even though one central unit controls all components.

Thanks to the focus on composability and compositionality, as RobMoSys does, a growing number of companies will be able to contribute software modules that can become commodities for the ecosystem, drastically lowering the entry barrier. This situation will only be reached if the underlying models are sufficiently fine-grained to allow the clear separation between “commodity” and “business sensitive” modules and models.

3.2.6 Agility

The speed with which robot users, in particular SMEs, can reconfigure their processes to a new run and thus avoid costly downtime is critical to maintaining profit margins. With the RobMoSys model-based technologies, robotic systems become easier to re-purpose. First, robot programming becomes easier, thanks to more intuitive user interfaces using well-know and easy to understand graphical languages such as behavioural trees. Second, RobMoSys enables far more flexible and faster reconfiguration not only of individual robots, but also of entire fleets and robot-related connected resources. While hardware aspects (such as different types of grippers) can play a large role in agility, they are not the focus of this project. A number of key technology enablers of innovation and value supports the RobMoSys approach, which are part of the software-defined manufacturing (SDM) landscape [SDM 2019], from virtualized flexible and hardware-independent facilities, over digital twin added value (RobMoSys Data Sheets), to fully dynamically self-adapting AI-based technologies (see Figure 6).

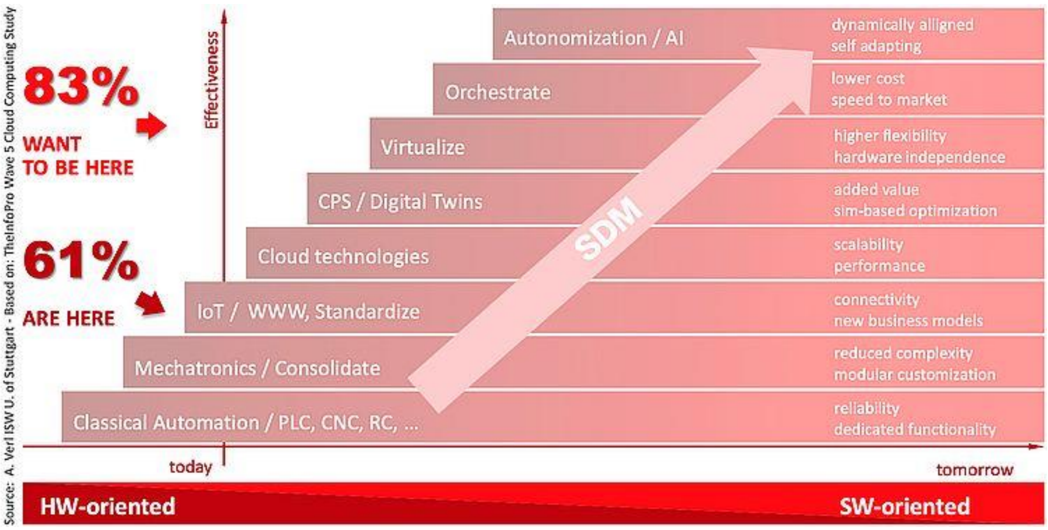

Figure 6. The software-defined manufacturing (SDM) landscape

3.3 Opportunities and Risks

RobMoSys strength comes from the strategy to secure adoption in the robotics community and sustainability and from the authoritativeness and complementarities of project partners, representing a guarantee of reliability and credibility. However, it cannot be ignored that the RobMoSys business success presents some risks. These are listed below, and they are shared with several partners' own business plans.

3.3.1 Unavailability of success stories

For new tools or approaches, in order to become affirmed or de-facto recognized, all domains of interest (manufacturing, agriculture, transport and logistics, etc.), are characterized by long latency. This might prevent applying the RobMoSys results to their full extent in industrial use cases before the end of the project. Therefore, here is a risk that RobMoSys remains a valuable research project only. Essential will be to establish and determine pilot customers or users, at least, and support them early on. The methodology will be demonstrated through pilot cases developed by third-party to further involve people external to the project, and get useful feedback. Last but not least, intensive dissemination activities are planned.

3.3.2 Inadequacy of use cases and requirements

Use cases are essential to demonstrate the business value of RobMoSys approach and tools. That means the use cases must meet business demands and shall not be artificial. Equally requirements that are derived from the identified business needs. It will be essential to have periodic reviews and collection of feedback. The strategy of RobMoSys is entirely devoted to secure adoption in the robotics community and sustainability. The situation today is that the robotics community has not developed yet common methods for software development, so that adoption of the RobMoSys approach could be an issue, mainly for developers. For this reason, RobMoSys has been built around a simple but effective concept: the involvement of the robotics and the software community. To that end, all the project partners' activities are massively focused on preparing appropriate specifications (concerted with Tier-1 experts) and running Open Calls.

3.3.3 Generic or complex results

With the ambition of meeting in one shot all robotics domains, the RobMoSys achievements might result too generic and complex at the same time, and thus scarcely acceptable by any single domain representative. Continuous advice by RobMoSys industrial partners is essential to mitigate this risk. RobMoSys intends to create as well synergies with other robotics platforms, as ROS and I-Cub, to further improve them. As for software, selection criteria in Open Calls will secure an open-source software and tools "track" for the eco-system and commoditization of basic building blocks of certifiable quality.

RobMoSys proposes a technical solution based on a **model-driven approach** that can be tailored to stakeholder specific needs, but practitioners could refuse more formal, not code-based approaches. Special information events, targeted campaigns and training will be undertaken. The project aims at showing the benefits of the approach since the very beginning through trials during the Tier-1 workshops especially targeted to software developers. Feedback will be taken into account to remove blocking points. RobMoSys will create related software projects in different platforms, such Eclipse, and web-based repositories, such as GitHub.

3.3.4 Difficult integration

The RobMoSys platform might turn out to be difficult to integrate with available tools and practices, unless spending time to import their know-how within this new platform and reducing its probabilities to be adopted. Periodic integration trials should be attempted to mitigate this. During the preparation of the specifications, to avoid imposing a quality standard unilaterally, representatives of industry and academia will be highly involved in the preparation of the smallest possible set of common concepts to share in the community as renowned best practices and methods. All the specifications will be publicly available, and the project's team activities constantly monitored by the community.

3.3.5 Visibility

RobMosys's visibility may be too low. RobMoSys proposes different instruments (Tier- 1, Open-Calls, RiF and CoC), that could cause some dispersion and loss of efficiency in the global communication. A clear communication plan including presentations at broad-spectrum and specific events will likely resolve this problem – just as some RobMoSys partners did very successfully within ECHORD and ECHORD++. Outreach to new potential communities is also secured by specific actions carried out by two foundations that hopefully will have a multiplier effect in their respective already established communities. The first list of members gathered for the Tier-1 group already witnesses the interest that the different communities may have in the project. Platforms and facilities will be chosen depending on the requirements and recommendations gathered during the project lifetime.

3.4 RobMoSys as a Core Infrastructure for an Agility-Driven Digital Platform

Current robotics systems do not employ advanced software engineering technologies to deliver innovative products with a high degree of flexibility. Rigid robot services, inflexible processes, and inability to easily (re)configure robotics tasks are the main limiting factors. RobMoSys aims at overcoming these limitations by focusing on model-driven technologies. An agile robotic system is a system that has the ability to accomplish service or production changeover between the execution of different robotic missions, with a minimum of change in tooling and program.

RobMoSys integrates state-of-the-art model-driven techniques to configure robots for arbitrary tasks, even by non-expert people, in an easy way. What is novel about the RobMoSys approach is that it does not only rely on software code, but also upon the perceived state of the world, known robot capabilities/skills and desired goals to follow a correct-by-construction development process, independent from specific middleware and programming technologies. The RobMoSys approach also supports model-based safety analyses by means of automated, constraint-based assessment of safety properties, including human-robot collaboration risks.

Model-driven development helps to virtualize the different engineering artefacts used to specify and deploy the robot and operator behaviour, their skills/capabilities, as well as software modules and implementation layers. This facilitates reuse of robotics assets in different contexts (different robots, robot versions, and upgrades) by means of dedicated configuration and instantiation tools.

3.5 Issues of using Open Source Software in the Machinery Industry

Despite the high value proposed when using open source software, one also has to admit that there are some issues related to the use. Especially from the machinery industry, that does not sell software as a standalone, but may incorporate open source software in their hardware products, serious concerns have been voiced.

Due to the high number of different open source licenses existing that include different rights and obligations (e.g. information obligations) for the user, the risk of inadvertently breaching terms of licences should not be denied.

Typical license requests are “publish on each copy an appropriate copyright notice and disclaimer of warranty” “and give any other recipients of the Program a copy of this License along with the Program”. In case of embedded devices and a multitude of open source software elements used, this is not an easy to fulfil requirement. In many cases, the effort for the manufacturer is tremendous.

Some standards for example request the user to provide the full source code with the software. In the usual case where the software has been modified to fulfil the application’s requirements, the manufacturer may see their process knowledge under threat when having to publish the complete source code. In many cases, the license text must be part of the contract – thus you would need to give access to the software before signature of the contract. In other cases, license text need to be included in the (embedded) device.

A significant number of cases have caused legal action and have been taken to court for copyright infringement, especially with GPL-2.0 licences. This can imply very high costs for the manufacturer (Injunctive relief, claim for damages, annihilation, recall from the supply chain, claim on reimbursement etc.) and lead to a complete stop of sales for the product concerned (i.e. a machine sold with the respective open source software).

To overcome those hurdles for the user in the machinery industry who is not so familiar with the use of open source software, the licence models chosen within the project should be kept simple, to a low number, and favour the use of business-friendly open source licenses. Maybe also a short guideline could be produced to make usage of software provided as easy and beneficial as possible for the user.

4 RobMoSys Community Business Model

An open source community is the keystone for the sustainability of the RobMoSys project. The RobMoSys project focused on creating an open robotics platform comprised of extensible frameworks, tools and libraries for building, deploying and managing software across the robot lifecycle. The main motivations for this community are:

- **Sustainability**: This ensures the sustainability of the RobMoSys platform artefacts. If the initial committers leave the project, the platform artefacts are available without any constraints to anyone, or any group, in the community who wants to continue to maintain and extend it.
- **Mutualisation**: By building a community around RobMoSys, we are able to mutualize our effort and resources. The community, including few of our resources, will maintain the core of the project while the rest of our team will work on our product, our added value, to make the difference with our "competitors".
- **Standardization**: RobMoSys defined a set of modelling concepts/patterns for the robotics domain (from low-level concepts such as communication patterns used in robotics middleware to high-level concepts such as Mission, Tasks, World Models, among others). At present, there is no modelling language with such a rich set of features specific to robotics that are, nevertheless, generic enough to be applicable to different domains (healthcare robotics, industrial robotics, service robots, etc.). These efforts provide strong seeds to create real, community-driven standards.
- **Metrics**: Last, but not least, a good way to know if the open source we want to use is sustainable and viable is to check the activity of its community: number of committers, number of commits, regularity of the release, activity on the mailing lists and forums, and the quality and quantity of assets built around the open source software is a great indicator. In other words, the community is an excellent evaluation metric for a project.

The goal of this section is to **identify the main elements of the RobMoSys business model**. This business model is focused on the open source **community strategy**. To this purpose, we adapted the Business Model canvas presented in Section 2, to open-source-oriented canvas as proposed by [Fachler 2016]. Figure 7 summarizes the RobMoSys Business Model, and the next sections elaborate each of the aspects of this business model.

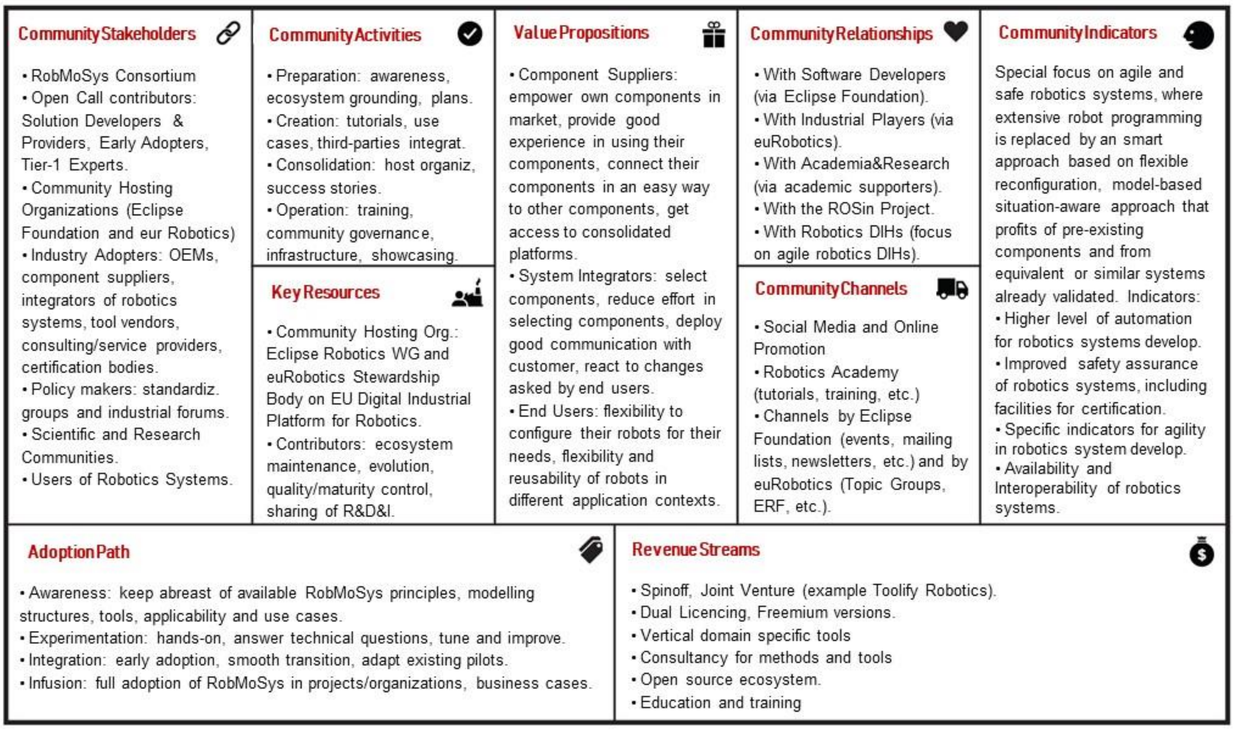

Figure 7. The RobMoSys Business Model summary

4.1 Community Stakeholders

The knowledge acquired in RobMoSys is expected to be strategic for European organisations. The RobMoSys stakeholders are shown in Table 1, and concern those actors who will be directly, or indirectly, positively affected by the RobMoSys ecosystem, its activities, and/or its results.

Table 1: RobMoSys Community Stakeholders

Target Group	Examples of stakeholders	Concrete examples
Industry: ▪ transport and logistics, ▪ manufacturing, ▪ healthcare, ▪ agriculture, ▪ other possible domains that RobMoSys could have an impact on.	OEMs, component suppliers, integrators of robotics systems, tool vendors, consulting and service providers, certification organizations, standardization groups and industrial forums. SMEs represent a special interest group for RobMoSys, as they generally have very limited access to basic or applied research to develop new products.	Magazino, Bosch Intralogistics, Bluebotics, Pick-It / Intermodalics, Robotnik, MIR, Swisslog, Omron Adept, Oceaneering AGV Systems, Elettric80, IEMA s.r.l., Cloos, Rio Tinto, Strauss Verpackungsmaschinen, INGRO Maquineria Hocoma, BA Healthcare, Focal Meditech, Endocontrol, F&P Robotics, Marsi-bionics, Instead Technologies AGCO/Fendt, Vitirover, Agrobot, DeLaval, Lely, Naio Technologies, Fullwood, GEA Farm Technologies qbrobotics, Robotnik, Blue Ocean Robotics BA Systemes, Force Dimension, InMach, MT Robot, Nilfisk, Intellibots, Adlatus Robotics Metralabs, RB3D, Schunk Pilz STENA recycling REFIND Technologies KUKA
Policy makers	Consultancy providers, assessor companies, standardization and certification bodies.	TÜV, Occupational insurance association (e.g. BGHM), ISO, European Mirror Committees euRobotics
Scientific and Research communities	Universities, research institutes, tech transfer organisations, and other practicioners.	Fraunhofer IPA, Fraunhofer IESE LAAS/CNRS University of Yourk DFKI

		Bristol Robotics Lab CHRU Scuola Superiore Sant'Anna Biorobotics Institute (agriculture) Leibniz-Institut für Agrartechnik und Bioökonomie Potsdam-Bornim e.V. Wageningen University & Research (agriculture) IK4Tekniker Research Alliance IRT Jules Verne Flanders Make (manufacturing industry)
Open-source communities	Stakeholders (development, quality assurance) of open-source ecosystems for Robotics.	ROS, Yarp, OROCOS OSGi MORSE (Simulation Engine)
Users of Robotics Systems	People that will ultimately use robotics systems or their services.	BMW, Audi, Bosch, Renault, Airbus

RobMoSys must create value for them by helping them overcome the constraints (e.g., costs, resources, maintenance, or reuse). **Erreur ! Source du renvoi introuvable.** provides a summary of needs per stakeholder type.

Table 2: Needs of RobMoSys Stakeholders

Stakeholders	Stakeholders Needs Targeted by RobMoSys
Original Equipment Manufacturers (OEMs)	• Enabling an efficient tool supported development process which suites the needs for robotic software and documentation in an optimal way. • Organizing the suppliers work related to the robotics products to achieve efficient supplier coordination. • Complying with the standards for development, assurance and certification.
Component Suppliers (Manufacturers)	• Specification of robotics modules, which can be integrated into the overarching robotic systems. • Transferring component artefacts (e.g. specifications, assurance results and documentation) across multiple domains. • Preserving the integrity of the components that they provide to platform integrators. • Ensuring the integrity of the components both up- and down-stream of the supply chain. • Application of tool support for the provision of composition of qualities and other properties so that integrated systems can be created with lower effort.
Integrators of Robotics Platforms	• RobMoSys results concerning the composition of the robotic software components based on individual modules, which

	ensure the integrity of the non-functional properties passed through the supply chain. • Tools that support these processes, with tutorials to educate adopters.
Consulting and Service Providers	• RobMoSys results which ensure the integrity of the components passed through the supply chain.
Certification Organizations	• RobMoSys results concerning intra/cross-domain and multi-concern assurance.
Tool Vendors	• Interoperability. The adaptation of the existing tools to the RobMoSys architecture and working philosophy. • Information access. Ensure that all the information relevant for the tool development is available.
Policy Makers and Standardisation Groups	• Practices for Robotics system and software development. • Model-based development practices.
Scientific and Research Communities	• Outcomes of the project. • Advance of the state of the art.
Open Source Communities	• An open tool platform developed by collaboration and with free redistribution and access to an end product design and implementation details.
Users of Robotics Systems	• Meet their requirements and constraints in terms of operation and maintenance.

Figure 8 shows stakeholders organized according to the level of contributions to the ecosystem during the project duration.

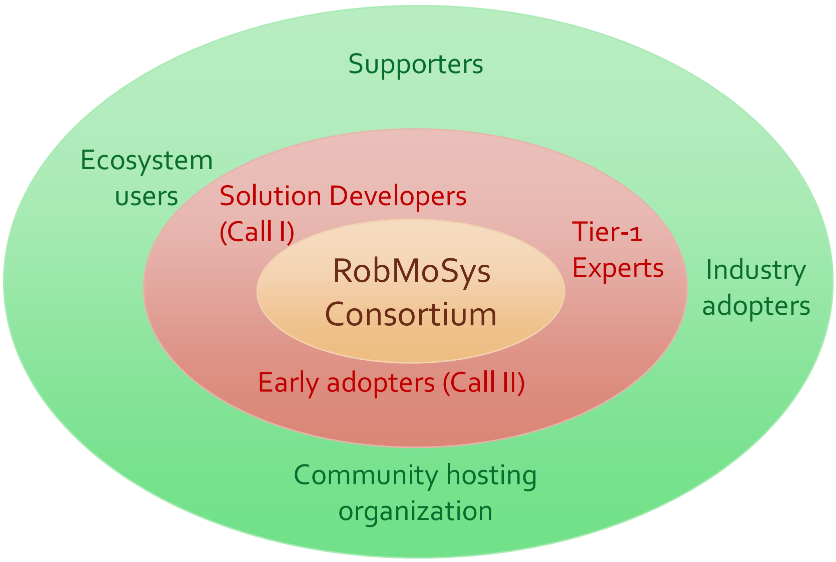

Figure 8: RobMoSys Community Stakeholders

RobMoSys consortium is at the core of this stakeholder view. The second-level contributors are Tier-1 experts and third-party organizations from Open Calls.

Tier-1 Experts Workshops allowed us to systematically gather requirements and recommendations helping project team members to (i) define the specifications of the Open Call, (ii) to monitor and assess the results of the Open Calls and (iii) to promote strategies for dissemination and exploitation. Workshops during the monitoring phase of Open Calls also hosted trial sessions with selected groups of users moderated by experts.

Third-Party Funding allows the "influx of knowledge and brains" from a broad set of stakeholders, so that the project was able to tap the tremendous potential of the community in a systematic way. In this way, RobMoSys was able to successfully "address the open development of integrated sets of tool chains and building block applications that support the construction of complex robotics systems". The Open Call mechanism allowed us to identify the best tools already available, the best modelers and developers to adjust them and the best application areas to validate the results and establish benchmarks.

Then in the third level, stakeholders include:

Supporters which are selected individually: industrial (non-funded) third parties, who helped the community to adopt the RobMoSys approach and models and code in the context of their company and the community. The motivations behind this means are (i) less risk for the adopting company, (ii) faster and more effective dissemination with more direct results, and (iii) more intense feedback for the RobMoSys partner from industrial reality.

Ecosystem users and industrial adopters. The change of industrial practices is a complex and long process. A single research project, like RobMoSys, cannot realistically expect to cause a revolution in industrial robotics processes in a short-term. However, the project is demonstrating a feasibility of the advocated approach and industrial RobMoSys partners, Tier-1 experts and third-party partners will be committed to internal dissemination of the outcomes and gradual adoption of both individual concepts and the entire framework developed by the project. It should be noted that one of the secondary, but very important, by-products of the project is establishing networks between industrial and research, training and consultancy stakeholders. Combined with exposure of industrial partners to cutting edge concepts and principles this naturally facilitates future smaller-scale collaboration that will, in turn, facilitates adoption of RobMoSys concepts in industrial practices.

4.2 Community Activities

A roadmap was created to plan the community creation activities, which divides the project into three phases: The preparation phase, the creation phase and the consolidation phase.

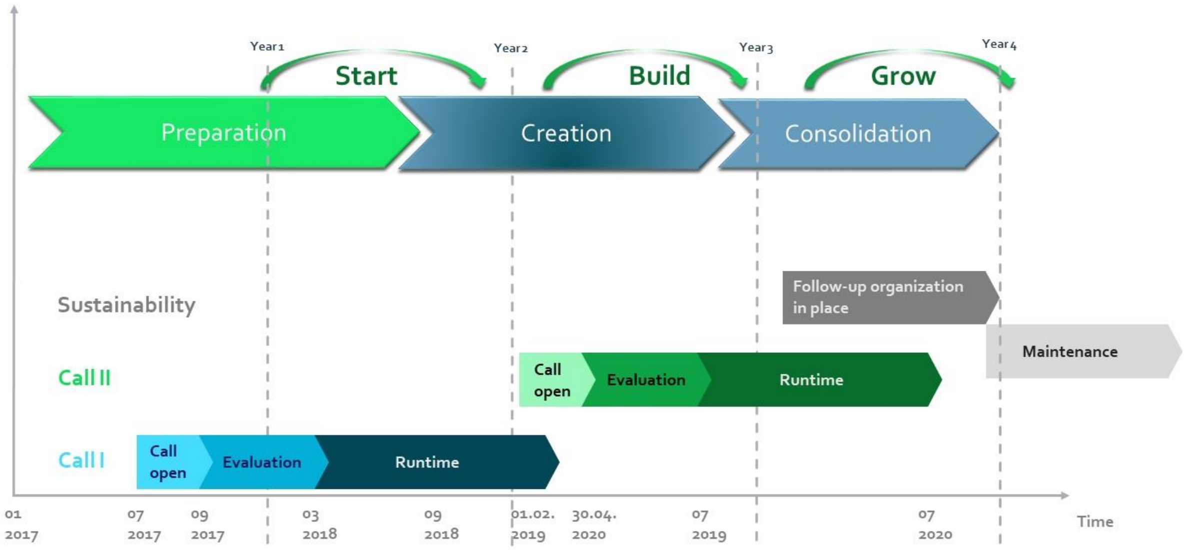

Figure 9: RobMoSys Community Phases

Figure 9, Figure 10 and Figure 11 show the three phases and their link to the two Open Call as well as to the other outreach activities: Dissemination, Training, Exploitation, Business Models and Community Building activities. During the preparation phase, the RobMoSys consortium focused on creating the two Eclipse projects: SmartMDSD and Papyrus4Robotics and providing the grounding for the sustainability strategy in RobMoSys. During the Creation phase, we produced the main substantial collective results of the project and bring them to the community by means of the communication infrastructure, videos, tutorials and usable tools and software. During the Consolidation phase, we focus in involving the host organization for the community in RobMoSys activities and strategy. We worked on the first success stories with industrial stakeholders and provided a strong basis to make the RobMoSys community a sustainable endeavour.

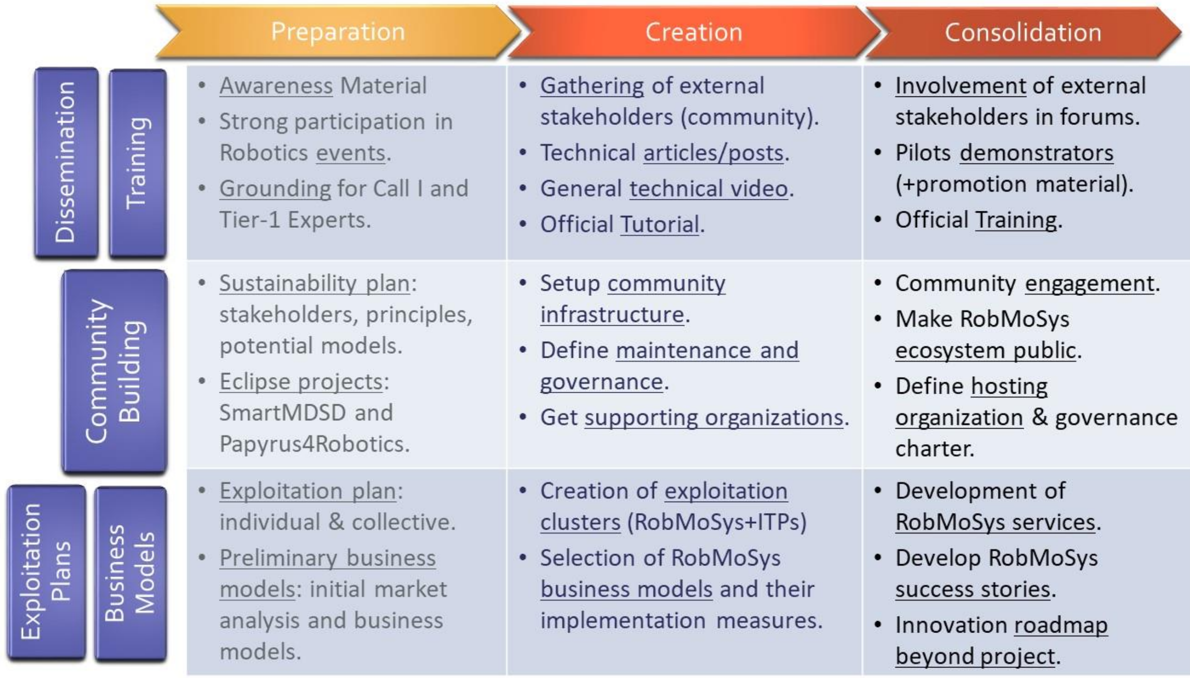

Figure 10: RobMoSys Community Macro Activities

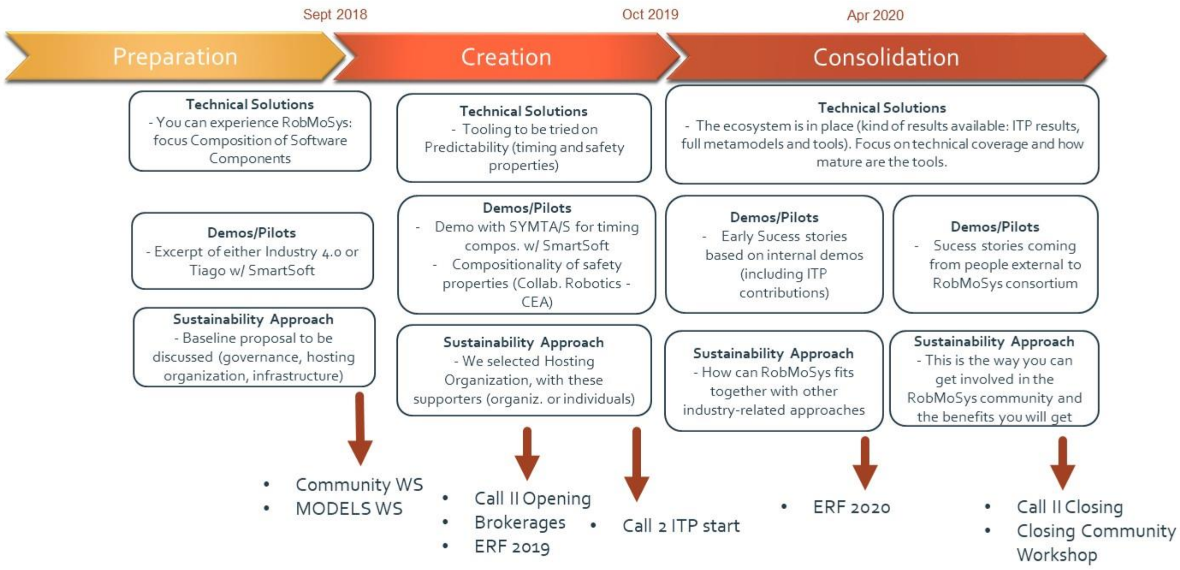

Figure 11: Strategy to Consolidate the RobMoSys Community

After the RobMoSys project end, the sustainability will be delegated to the Stewardship Body on EU Digital Industrial Platform for Robotics as an instrument within euRobotics and the Eclipse Robotics Working Group. This is described in detail in deliverable D7.9 (Sustainability Plan). The main immediate activities related to that sustainability plan are:

(1) Setting-up the community governance and collaboration model among all stakeholders of those working groups. This includes the Charter publication and setting up the operative tools and procedures needed to run them.
(2) Publishing the main RobMoSys communication infrastructure, including the website for RobMoSys Academy and setting any supportive tools to facilitate the work of the community contributors.
(3) Showcasing the impact for sustainability. At this stage, our action will be dedicated to showcase the results and use them to engage with future members and sponsors to contribute to the RobMoSys

Platform sustainability.

4.3 Key Resources

RobMoSys built an open sustainable ecosystem around public, royalty-free and implementation-driven software platform standards that ease the development of new robotic applications in multiple domains. The innovation required to advance RobMoSys methodology and tools needs to be driven by the key industrial companies. This is why RobMoSys partners selected both euRobotics and Eclipse Foundation to host the RobMoSys results, euRobotics from the conceptual point of view and Eclipse from the technological solution perspective. These community hosting organizations guarantee a suitable environment for open innovation and industrial feedback.

RobMoSys partners expect that the project solutions will evolve in pace with the more challenging requirements of modern robotics engineering teams. RobMoSys solutions needs flexible extensibility and customization that makes it easier to adopt its tools, methods and processes by robotics engineering teams.

The main resources of the RobMoSys community will be both the hosting organizations and the contributors to the RobMoSys ecosystem.

4.3.1 Role of the Community Hosting Organization

The goals of the community host organizations are:

- Providing means of collaboration between RobMoSys contributors and other stakeholders.
- Organizing sustainable commercial services and ecosystems around open source models, software components and tools.
- Fostering exchanges between academics and industry partners.
- Managing the quality and maturity of the concepts, tools and components from early research prototypes through to obsolescence.
- Providing the documents and qualification kits required for conformace with the RobMoSys development standards.
- Recognizing project maturity and company know-how and commitment through a branding process for conformance to RobMoSys.
- Ensure long-term longevity of models, software components, and tools since they must last for a long time.

The idea of joining an existing community is to avoid re-developing community management aspects but also to connect with other software communities. Many very good solutions answering some industrial needs already exist in open innovation communities. But most of the time, specific issues like durability or conformance branding are not taken into account. In this case, the community platform must play its part by providing tool assets, setting up specific support, and coordinating development and support.

4.3.2 Role of RobMoSys Contributors

The role of RobMoSys contributors in such a community would be:

- Preparing RobMoSys models, software components and tools to be released/hosted in open source.
- Operating dedicated code repositories, build chains, test facilities, etc.
- Fostering exchanges between RobMoSys partners and industry partners.
- Proposing RobMoSys tool enhancements (industry-friendly functionalities, new features, reliability features, tool connectors with other external tools, among others).
- Managing the quality and maturity of RobMoSys tools
- Ensuring open innovation through the sharing of the research, development, and maintenance efforts as far as possible
- Fostering sustainable commercial services and ecosystems around the RobMoSys tools.

The main goal is to use RobMoSys partner’s technical expertise in the models, software components and tool platform and comprehensive understanding of the ecosystem challenges, in an effective way by offering continuous support for industrial players wishing to use these technologies in a cost-effective way for long-term projects.

4.4 Value Proposition

In order to define the RobMoSys business models, we simplified the users in three representative groups: component suppliers, system integrators and end customers.

4.4.1 Component Suppliers

Component suppliers are responsible for assuring the critical properties of their delivered products. Component suppliers need to support higher-level integrators in their assurance and certification processes.

Component suppliers are mainly concerned with quality, time to market and cost efficiency and reliability, when developing robotics software components. The main difficulties and challenges faced by these users are related to controlling the baseline technologies they use, describe the functionalities they provide in their components in a user-friendly way, too many different interfaces to be supported at the same time, the unavailability of a software component market and poor software tools. They expect to empower their own components in the market, provide good experience in using their components, connect their components in an easy way to other components, get access to consolidated platforms to develop their components.

The resulting value proposition map for the component suppliers can be seen in Figure 12.

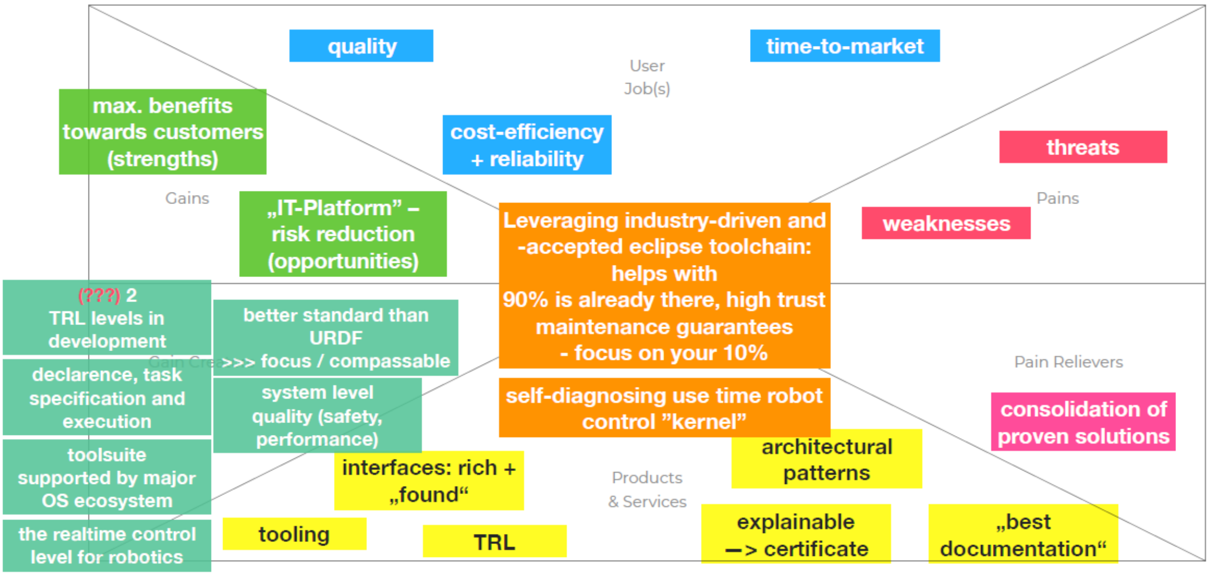

Figure 12: Value Map for Component Suppliers

4.4.2 System Integrators

System integrators refers to the manufacturer that integrates the parts assembled and installed during the construction of a new manufactured product. Within the industrial setting, platform integrators are ultimately responsible for the dependability (e.g. safety) of the products delivered to the end users of the consumer market. "Composability" is a daily concern for system integrators, so they are a key source of adopters to target and educate.

System integrators need to understand what end users want and be able to react to changes cost-effectively/economically. This includes to select needed components, reduce effort in selecting components and compositions of the systems, to deploy good communication with customer, react to changes asked by end users during development. The main difficulties they experience are related to increasing the demand of flexibility at no extra costs, managing the change in market trends („legacy"), getting components ready for the certification process, managing customer demand of last-minute changes „for free", dealing with difficulty to open new application domains/markets for robots (economically).

The resulting value proposition map for the component suppliers can be seen in Figure 13.

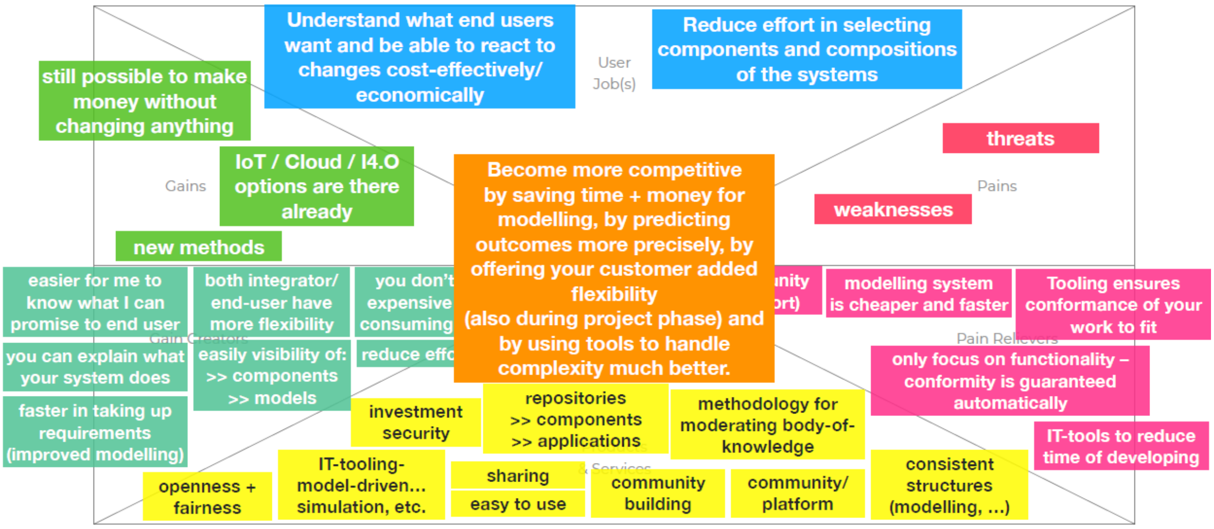

Figure 13: Value Map for System Integrators

4.4.3 End customers

End customers of robotics systems are the people that will ultimately use the robotic systems and their services. They are the people who will work most closely with integrators to generate results. They are interesting and valuable for RobMoSys from the robotic system operation and maintenance process viewpoint. These users can help identify more real dissatisfaction than almost anyone else in the robotic system production and operation chain. RobMoSys should approach them to understand their needs and what the RobMoSys ecosystem needs to do for them in order to get results.

End customers look for flexibility to configure their robots for their needs, flexibility and reusability of robots in different application contexts. They also look for reliability, by keeping their systems, making no mistakes and by staying safe at operation. Finally they look for competitiveness (be better than competitors, keep robots cheap, stay independent of specific vendors). The challenges they face are related to lack of sustainability of product life cycle, lack of support and stability of robotic systems and lack of options for easy reusability, lack of flexibility to manage various robots variants operating in multiple contexts.

The resulting value proposition map for the component suppliers can be seen in Figure 14.

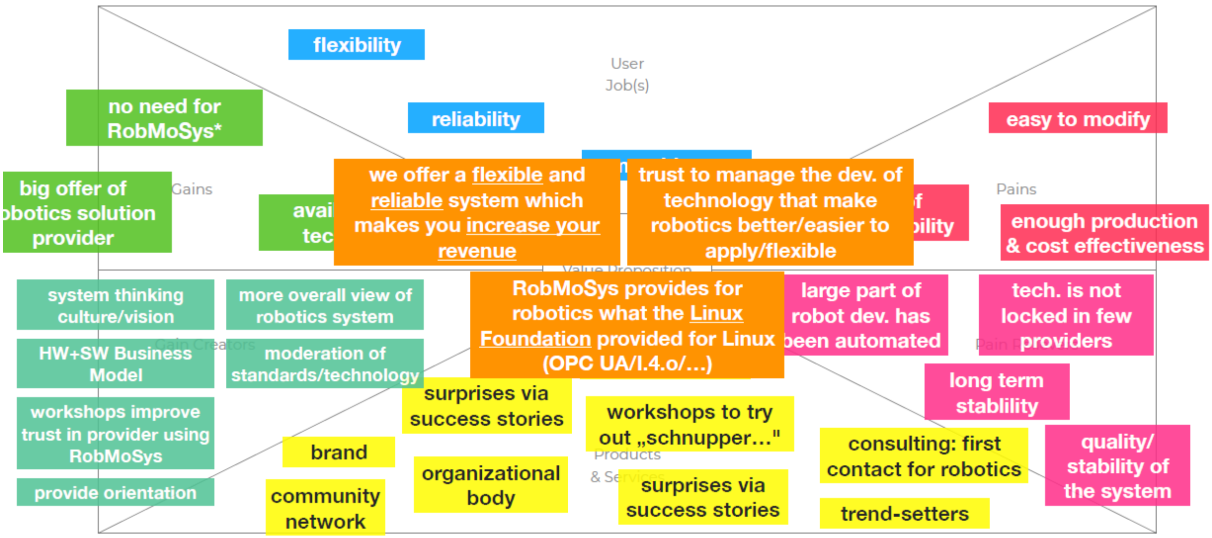

Figure 14: Value Map for End Customers

As a result of this study, we got the following preliminary conclusions regarding the best target groups for RobMoSys:

- This group does not have to be robotics costumer
- All costumer in need of automation solutions, or in need of advanced solutions
- We should target groups responsible for R&D as a first approach.
- They should have familiarity with how to make money and risk minimization with open source.
- Size of companies or teams about 25 or more.
- We should target R&D managers who make the “early adaptor“ decision.

4.5 Community Relationships

4.5.1 Relationship with software developers

RobMoSys needed to generate the acceptance of software developers of different application domains (automotive, aerospace, etc.) within and outside of industrial companies: Within the consortium, this target group has been represented by the Eclipse foundation and will continue to be supported after the end of the RobMoSys project. Eclipse is highly familiar with the information requirements of this target group and has been able to identify and produce the online media to which this target group has been very responsive, in particular:

- Users of existing digital platforms, i.e. communities with specialised customer groups interested in a specific technology or domain (e.g. the DDS middleware community)
- Users of robotics middleware, robot simulators and representatives of educations robotics (high-profile representatives are members of the Tier-1 group)
- Developer Communities, like the global Eclipse open source community that involves thousands of developers worldwide, where software developers could discover and adopt RobMoSys tools and methods both for robotics activities and for the development of cyber- physical systems.

4.5.2 Relationship with industrial players

In the first half of the project runtime, EUnited was a member of the consortium and responsible for paving the way of RobMoSys to decision makers in industry, the hierarchy level that needed to be addressed depending on the size of the company (large industry, mid-caps, SMEs). In 2020, PAL robotics took over this role. This will continue after the end of the RobMoSys project by means of euRobotics (PAL Robotics is member of the industrial board). The target groups are:

- Representatives of highly influential industrial domains like automotive, aerospace, cleaning (again: powerful representatives are part of the Tier-1 groups) – they could be key enablers for the uptake of the platforms for digitalization of robotic systems
- The target groups addressed by the RobMoSys Open Calls (mainly also SMEs): tool makers, system integrators, modelers, component suppliers
- Representatives of the entire value chain.

4.5.3 Relationship with academia and research

It is important that the RobMoSys’ outcomes are considered and perceived as community effort based on broadest possible involvement of expertise from academia and research. Based on this, it was also decisive that very early renowned institutions in academia and research deploy the RobMoSys outcomes in their environments. KUL, TUM and THU with their tight networks and strong links into different Topic Groups of euRobotics including exploitation of the Tier-1 group have put strong effort into making all the places offering RobMoSys in teaching and research visible as a strong and growing “places-to-be”. CEA is also member of the research board of euRobotics. The target groups are:

- Colleagues in robotics (teaching as well as research) that advise their researchers and PhD students to base their work on RobMoSys outcomes;
- Wider outreach by approaching colleagues in all related domains, such as software engineering etc., to form a growing number of multipliers and to attract the best brains to grow the RobMoSys ecosystem (including PhD schools, summer / winter schools);

- Including the methodology developed in RobMoSys in the curricula will enable industry to hire well trained applicants, easing the uptake of RobMoSys outcome in their research and development activities.

4.5.4 Relationship with the H2020 ROSin project

The European Commission decided to double the funding for innovation actions on “system development tools” in the call “H2020 ICT-26: System abilities, development and pilot installations”. With the resulting high expectation towards the success of these two closely related projects, the EC voiced the need for closer collaboration between these projects. In a first joint RobMoSys-ROSIN project meeting (initiated by the EC in February 2017), the slogan “EU Digital Industrial Platform for Robotics” was developed as a joint header for both projects. The slogan has been used to market this approach. The concept was further shaped during the runtime of the project.

We regularly invited the ROSin project to participate in our ERF workshop as a speaker with a presentation of their project and offering them to chair a table of the World Café. The European Commission scheduled joint workshops accompanying the review meetings in their premises in Luxembourg. Both projects invited other relevant EU-funded projects to share best practises, develop joint concepts and share thoughts about possible future collaboration and mutual benefits. The coordinator of the ROSin project has also been invited to participate as a panellist at our Conference and closing event in January 2021.

This relation with ROSin will be now consolidated by means of the creation of the Robotics WG at Eclipse Foundation, where both CEA (RobMoSys) and Fraunhofer IPA (ROSin) will lead the working group and will work together in consolidating both digital platforms.

4.5.5 Relationship with Robotics DIHs

RobMoSys conceptual outcomes seek to increase their maturity and establish a strong, coherent and consistent basis for industrial uptake both by robotics developers, integrators, support tool vendors and consulting companies. In addition, another objective is to positively influence and help the various standardisation bodies towards a more rigorous interoperability of manufacturing related standards by adopting concepts coming from RobMoSys such as the Data Sheets for industrial robotics. In addition, cooperation with existing DIH networks, such as for example the agile manufacturing-oriented ones – DIH^2 and TRINITY as well as the RIMA DIH network on Inspection & Maintenance, coordinated by CEA – is crucial to the success of the RobMoSys project. (The COVID crisis turned out to be a major problem for the dissemination of RobMoSys concepts and results within the DIH networks, because of the importance of personal interactions and customized explanations.)

4.6 Community Channels

One of the key aspects for a successful RobMoSys community is the necessity of proper communication tools. Different communication channels needed to be addressed in order to reach out to all target groups. Some of them have been important from the first day of the project, others will become more important after the end of the RobMoSys project.

4.6.1 Social Media and online promotion

Social Media accounts have been set up for the project, with the aim to foster more familiar communication. The icons have been integrated in the website, for the audience to be aware of the availability for “conversation” of the project. Social Media announcements and personal contacts turned out to be even more important to create understanding and involvement among the important stakeholder groups (current and potential users of the model-driven software framework) than classic media coverage.

This includes regular press releases and newsletters, all planned events (conferences, workshops, etc.), social media and online promotion as well as training activities (“summer schools”). A first set of communication channels, like press releases, website, wiki, social media channels (twitter, YouTube, LinkedIn) as well as open community workshops and brokerage days have been established throughout the project.

4.6.2 RobMoSys Academy

During the final reporting period, we launched the RobMoSys Academy which initially only included a link to the RobMoSys Wiki, our YouTube channel and the Pilots' site. We invested a considerable amount of time during the COVID19-pandemic to produce six tutorial videos, explaining the main goals and advantages of the RobMoSys approach.

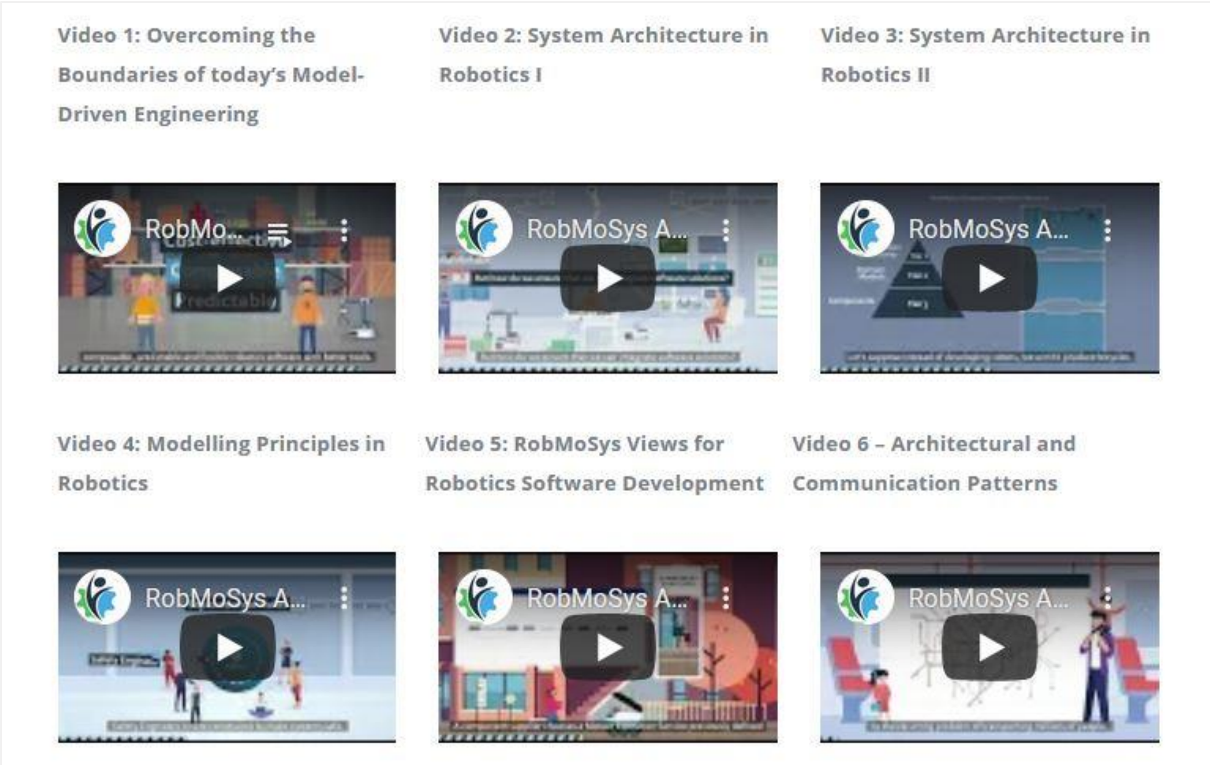

Figure 15: Screenshot of the RobMoSys Academy tutorials

Its content is structured in three main areas: Methodology (Wiki), Videos and Pilots. Six consecutive videos explain the RobMoSys approach in a simple way :

1. Overcoming the boundaries of today's Model Driven Engineering
2. System Architecture in Robotics I
3. System Architecture in Robotics II
4. Modelling Principles in Robotics
5. RobMoSys Views for Robotics Software Development
6. Architectural and Communication Patterns

After the RobMoSys project, training must be reinforced. It is important to achieve a high impact by choosing the appropriate training methods. This includes, as far as possible, a train-the-trainer model, which enables experienced personnel to show a less-experienced instructor how to deliver courses, workshops and seminars. While part of the delivery of the training programme was and will be carried out by traditional approaches (such as "class-based" courses, seminars, and workshops), it is important to develop and establish new ways of teaching, as time constraints are becoming more and more restricted for industry professionals. KULeuven included MDSD in their lectures. And also TUM conducted a Lab Course Model-Drive Approach for Robotics Perception for Master students, where the students learned how to use the tool RobMoSys tool Papyrus4Robotics to implement in a composable and modular way state-of-the-art perception algorithms. This must be reinforced in the upcoming months.

4.7 Community Indicators

Despite the wide adoption of the concepts of model-based development, their entrance in robotics domains such as collaborative robotics and areas like manufacturing, is not advancing at the pace that the designers and producers would want in order to exploit the many benefits brought to these applications.

The RobMoSys project brings a new vision into agile robotics where extensive robot programming models are replaced by an intelligent approach based on flexible reconfiguration, model-based situation-aware approach that will profit not only from previous project results of pre-existing modules but also from equivalent or similar systems already validated.

This process of automated software design and validation from similar architectures is performed more or less unconsciously by all the designers during early architectural design phases. All the designers and companies rely on a series of architectures that are well known "to work properly". The RobMoSys project provides a systematic methodology and tooling to pass from this qualitative and intuitive approach into a standard-based, cognitive procedure where the underlying robotics architecture is specified in architectural patterns and reused across projects. RobMoSys extends this approach to architectures with inherent dependability properties such as safety.

The RobMoSys digital ecosystem is a key element in the impact strategy, as it will reduce dramatically the entry barriers of new actors, such as SMEs manufacturing customized products, in the production business by providing them with a consistent and easy-to-use task configuration environment that shall reduce their learning curves and increase their chances to perform a "right-first-time" automation of industrial robotic tasks.

To obtain the maximum impact from this new approach it is necessary that the proposed methodologies and software assets are perfectly aligned with both the industrial manufacturing processes and standards, and with the emerging factory architectures derived from cutting edge Industry 4.0 systems. Here is where the full potential of RobMoSys will develop. Due to the careful design of the RobMoSys composition structures, RobMoSys users get access to the most commonly used robotics system architectures, adapting the existing middleware layers, software libraries, tools and procedures to them and therefore guaranteeing the applicability of the results in the robotics industry.

Table 4 summarizes the main RobMoSys ecosystem indicators, including progress indicators, that should be considered after the end of the RobMoSys project.

Table 3: Main RobMoSys Ecosystem Indicators after the RobMoSys project end.

Aspect	Description	Progress Indicators
Higher level of automation for dependable Avairobotics development activities	Development of metrics to evaluate improvements in the level of automation of robotics development activities; development of effective techniques and tools that provide automated assistance in robotics development activities, such as robot orchestration, configuration, planning, code generation and design of safe by design mechanisms.	• Number of evaluation metrics developed (X). We anticipate X > 10. • Usefulness, representativeness, clarity, measurability, and objectivity of the developed metrics. • Percentage of robotics development activities covered in tool support (Y). We anticipate Y ≈ 60%. • Empirical evaluation of improvements in automation based on the developed metrics.
Improved Safety Assurance of architectural concerns	Development of metrics to define robotics scenario coverage (e.g. graph-based metrics for use case models) and safety requirements coverage. Development of metrics to define fault severity, e.g. for injected faults.	• Increase in percentage of fault-tolerant, fail-safe and fail operational robotics behaviours over the entire scenario range. • Decrease in percentage of safety violations. • Increase in the level of robotics tolerated fault severity while still exhibiting fail-safe and fail operational behaviours. • Decrease in percentage of total robotics system failure.
Dedicated focus to Robotics in Agile Production	Identifying and classifying RobMoSys usage scenarios and business constraints (budget, schedule, required trustworthiness margins, processes) with specific	• Number of identified scenarios (X) and constraints (Y). We anticipate X > 2 and Y > 10 for each survey or field study conducted, with potential overlaps between the scenarios and constraints in different

	focus on robotics components; practitioner-oriented guidelines targeted at the stakeholders of the agile robotics value chain describing how to use RobMoSys results; involvement of all directly relevant robotics development stakeholders	surveys and field studies. • Number of guidelines (Z). This number depend on the extent of the material and how it is decomposed, but we expect Z > 5. • Breadth of guidelines in terms of end-user needs covered. We plan to cover all major stakeholder roles, i.e., robotics manufacturer, component supplier, and tool vendor. • Level of stakeholder participation and feedback.
Incensement of availability and interoperability concurrently	Development of a ground-breaking methods for robotic application deployment and management of DataSheet-based interoperability at runtime. Consideration of uncertainties that an autonomous system must face. Building a safety baseline on which optimization of the system can be performed without harming a robotic systems dependability.	• X percentage of safely processable situations can be handled in an optimized way • Decreased number of false positives (X) in comparison to nowadays systems are expected, leading to higher availability of a system. We expect X to be decreased by up to 30%. • Optimization of performance of a robotics system in Z preciously non-dependable context situations. We expect Z > 10.

4.8 Adoption Path

The definition of the RobMoSys adoption process is key to get them on board.

RobMoSys defines a process for stepwise increased adoption levels of the RobMoSys approach and community interaction (Figure 16).

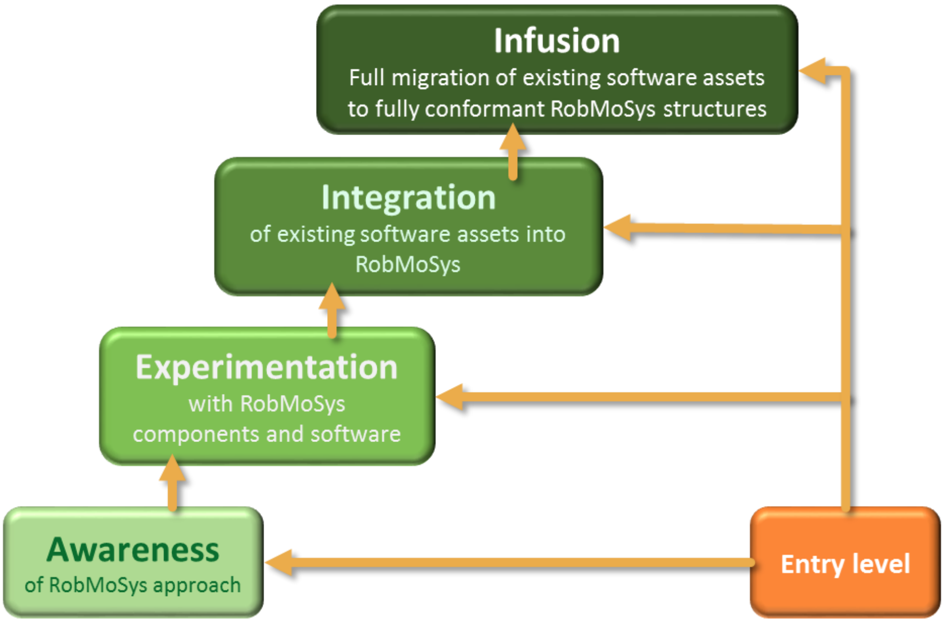

Figure 16: The RobMoSys Adoption Path

Level 1: Awareness. This is an important precondition for adoption. RobMoSys provides a structured Tutorial, as well as User Stories (https://robmosys.eu/user-stories/). Other important RobMoSys awareness means are newsletters, Brokerage Days, workshops organized by RobMoSys partners, and Discourse Forum. The main goals of this level are to:

- keep abreast of available RobMoSys principles, modelling structures and tools,

- understand applicability and limitations of the RobMoSys approach to the development of robotics software,
- actively seek to apply RobMoSys to appropriate problems, and
- show willingness to adopt the RobMoSys approach and technologies.

Level 2: Experimentation. It implies to setup and run experimental cases to understand and test the RobMoSys approach. RobMoSys provides two toolchains with User Manuals and Usage Scenarios to be reproduced. In addition, a set of RobMoSys pilot skeletons are available to work on real-world case studies. Finally, RobMoSys fosters "internships": motivated people can spend some time in RobMoSys partner labs, to get embedded in the RobMoSys approach, and to learn first-hand from the core developers. The main goals of this level are to:

- get hands-on understanding of the RobMoSys approach,
- answer technical questions and hypothesis by conducting controlled experiments,
- identify any technical constraint to apply RobMoSys in real-world cases, and
- improve, fine tune and extend all RobMoSys information (tutorials, wiki,...). Not in the least by adding to a repository of "best practice designs" of concrete robotic systems.

Level 3: Integration. This is a first step of the RobMoSys migration path. It implies the usage of RobMoSys technologies (models, software components and tools) by robotics development users. These users may keep their existing assets, by using connecting mechanisms such as the RobMoSys Mixed Port Component, or partial conformance to RobMoSys structures. The main goals of this level are to:

- start by an early adoption of RobMoSys approach, using RobMoSys architectural patterns and associated tooling,
- support smooth transition to full RobMoSys benefits (compositionality, predictability), by still reusing existing components and systems, and
- develop or adapt (existing) pilots demonstrating RobMoSys added value in the context of real industrial settings.

Level 4: Infusion. This step implies the full migration of existing assets to fully conformant RobMoSys structures. The main goals are to:

- show full adoption of the RobMoSys approach in an organization,
- demonstrate complete business cases showing a clear Return of Investment (RoI), and
- understand pros and cons on how RobMoSys permeates (an area of) an organization.

The advantage of these different levels (different entry levels with different support from our side) is that we can produce win-win situations at various levels of involvement: migration pilots, coaching, expert advice, and incremental adoption.

4.9 Revenue Streams

In particular, after the end of the project, we expect that RobMoSys partners apply different business model archetypes to exploit these items, which are classified and explained next:

1. **Spinoff / Joint Venture**: in order for some partners to further enhance their products and technologies, a spinoff could further enhance the outcomes of some of the product results, supporting direct marketability. An example of Spin-off created from RobMoSys is Toolify Robotics (www.toolify.eu).
2. **Dual licensing**: The open-source licenses do not strip the developers from their IPR rights, which they can license again through other conditions. RobMoSys partners (e.g. CEA, HSU) may be willing to sell extended licenses of their own products, adding support or management services, including warranties or liabilities.
3. **Freemium versions**: Likewise, the same company which creates an open-source (free) tool may create extended (premium) versions, which include other features, and which are only provided under commercial licenses. RobMoSys partners (e.g. CEA, HSU, TUM) may use this model to provide commercial versions.

4. **Vertical domain specific tools:** As a special case of the previous, RobMoSys partners (e.g. SIE, PAL Robotics, COMAU) will leverage the proprietary knowhow raised from different domains of application to create these tools; the same idea will be extended to other domains in the future.
5. **Contracting of tailored extensions**: Organizations that wish to use specific versions of the methods and tools that match their need may resort to using versions for their industry (just explained), develop their own extensions (later explained) or contract RobMoSys partners for their development.
6. **Consultancy for methods and tools**: Even though organizations can apply the method for free, it may not be worth it for them to acquire the necessary expertise (paying and or waiting for it), and they may rather resort to sub-contracting experts. RobMoSys partners (e.g. CEA, TUM, PAL Robotics), as the creators of these methods and tools, are best positioned to apply this model.
7. **Procurement** differentiators: purchasers following specific regulatory requirements (for cybersecurity, security, safety, specific market related – COVID-19 – requirements) having as part of the European guidelines.
8. **Open-source ecosystem**: The development of open-source projects requires both technical and institutional support to flourish and develop (Eclipse Foundation expertise). The former is supported by a web-based collaborative software infrastructure (forge) and the latter is provided by a community that oversees the development processes and is endowed with an appropriate governance infrastructure. Members and sponsors of the community fund the maintenance of these technical and institutional support mechanisms
9. **Education and training:** Experts in the methods and tools may provide education and training services to organizations willing to apply and use them. In particular, the project academic partners (TUM, KU Leuven) expect to apply this kind of exploitation, both within their regular courses and with in-house training solutions (potentially in association with industrial RobMoSys partners).

The running and management of the RobMoSys not-for-profit organizations will require minimum resources per year. The following assumptions regarding the sources of income that the project have been made to plan for sustainability:

- Budget from EU Projects.
- Sponsorship from corporates and regions to continue the operations of the Eclipse Working Group.
- Membership fee of the Eclipse Robotics WG.
- Transaction based fee for:
    - The marketplace for integrators or developers selling their technology.
    - Tailored consulting or training on the use of the technologies

5 References

[Osterwalder 2010] Alexander Osterwalder, Yves Pigneur, "Business Model Generation", John Wiley and sons, 2010.

[OpenSource Web] http://www.opensource.org/

[FSF Web] http://www.fsf.org

[Roadmap 2015] Robotics 2020, Multi-Annual Roadmap For Robotics in Europe, Call 2 ICT24 (2015) Horizon 2020, 2015, https://www.eu-robotics.net/cms/upload/downloads/ppp-documents/Multi-Annual_Roadmap2020_ICT-24_Rev_B_full.pdf

[Technavio 2016] Technavio, "Global Robotic Software Platforms Market 2016-2020", 2016.

[Girard 2016] A. Girard and C. Rommel. The Global Market for Software and System Modeling Tools, 2016. IoT & Embedded Technology. http://www.vdcresearch.com/Landing/iot1/A-15-Software-Systems-Modeling-Tools.aspx?bms.tk=BzAEqwsJp20Hm21Vr30Sl33Ss26Bk17Ms20BvfrFtg

[Eclipse 2016] Eclipse Foundation. Developing an open source MBE tool suite based on the Papyrus platform, 2016. http://www.polarsys.org/ic/papyrus

[OPEES 2013] OPEES Project, “OPEES Deliverable D1.2.1 - Open-Source Business Models for Polarsys”, 2013.

[SPARC 2013] The SPARC Partnership for Robotics in Europe, “Strategic Research Agenda for Robotics in Europe 2014-2020”, 2013.

[IFR-WorldRobotics 2017] Interantional Federation of Robotics, IFR Forecast 2017.

[Murphy 2017] Andrew Murphy, “ROBOTICS SOFTWARE AND SERVICES IS WHERE THE TRUE VALUE LIES”, 2017.

[SDM 2019] https://ifr.org/post/managing-mass-customisation-with-software-defined-manufacturing

[Fachler 2016] https://opensource.com/article/16/12/open-source-canvas