RobMoSys

H2020—ICT—732410

RobMoSys

Composable Models and Software for Robotics Systems

Deliverable D7.5
Business Models for the Ecosystem – M30

Huáscar Espinoza (CEA), Gaël Blondelle, Fariba Khatami (EUR), 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: D7.5

Document title: Business Models for the Ecosystem – M30

Version: 1.0

Due date: June 30th, 2019

Delivery date: 30.06.2019

Nature: Report (R)

Dissemination level: Public (PU)

Editor: Huáscar Espinoza (CEA)

Author(s): Huáscar Espinoza (CEA), Gaël Blondelle, Fariba Khatami (EUR), Anna Principato (TUM)

Reviewer: Ansgar Radermacher (CEA)

Executive Summary

This deliverable represents the second iteration of Business Models for the RobMoSys Ecosystem (Deliverable D7.5), 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. Identifying the RobMoSys user profiles and the value propositions for them. We plan to use this information to discuss with relevant RobMoSys stakeholders and experts during the last period of the project.

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 this 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.

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 strengthen. 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 will be incrementally specified and continuously aligned to market needs. This deliverable represents the second iteration of **Business Models for the RobMoSys Ecosystem (Deliverable D7.5)**, 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 this 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.

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 preliminary 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 some internal studies to prepare the business models aligned to the market needs and RobMoSys approach.

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 8 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).

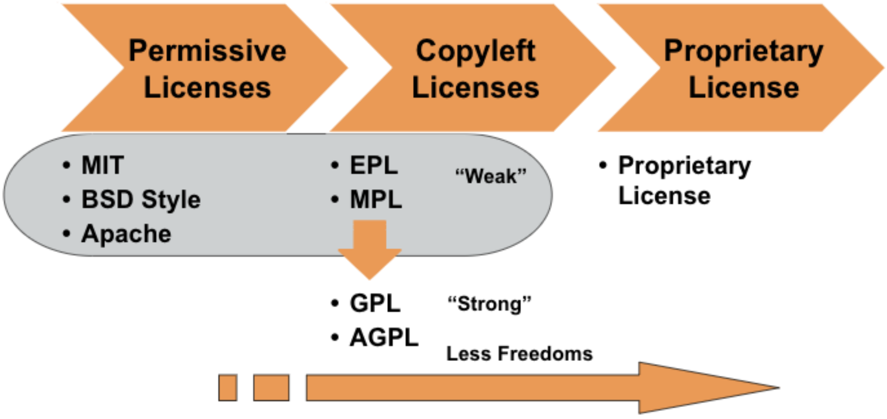


Figure 1: The License Spectrum

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.

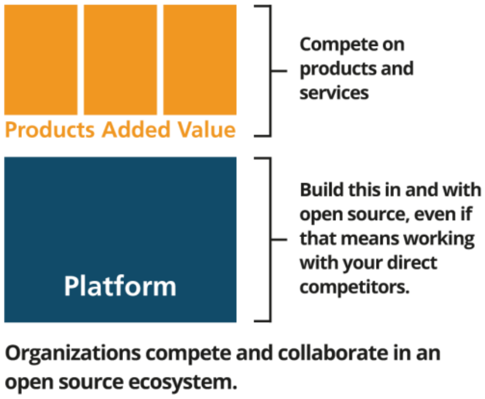

Products Added Value
Compete on products and services
Platform
Build this in and with open source, even if that means working with your direct competitors.
Organizations compete and collaborate in an open source ecosystem.

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.

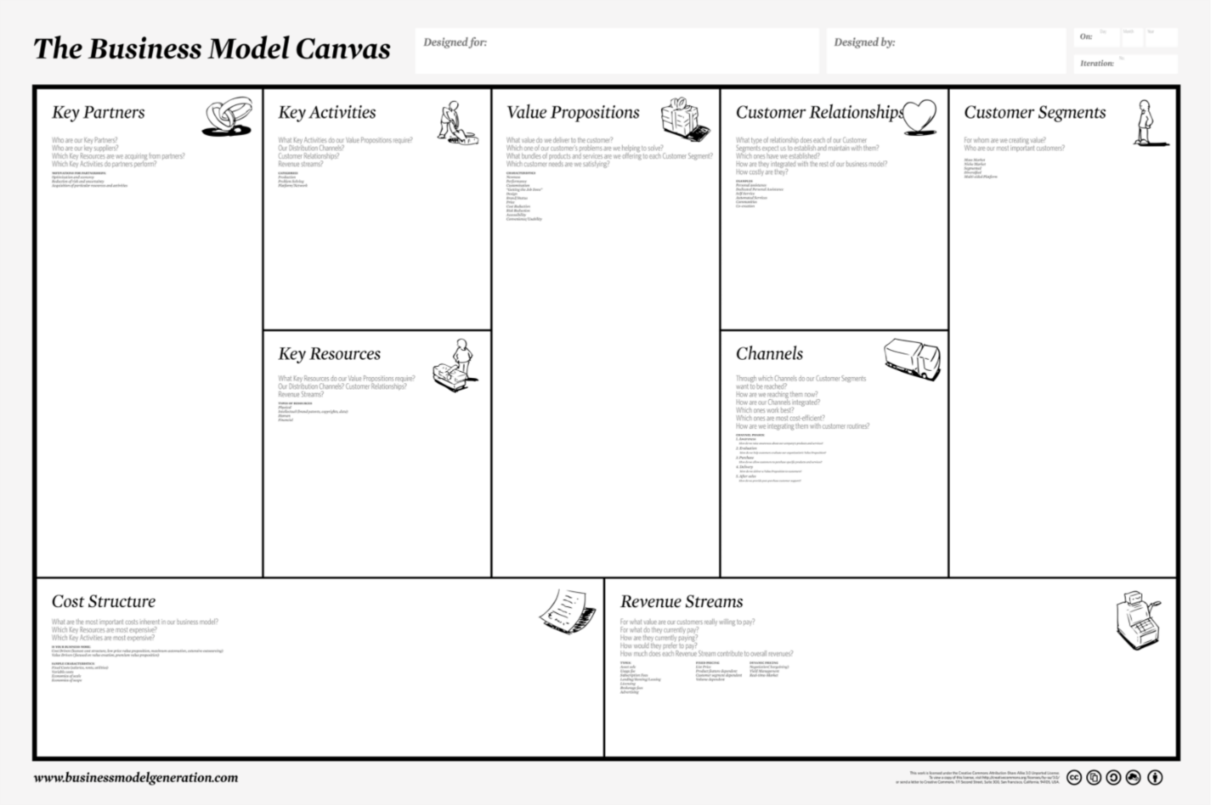

The Business Model Canvas
Designed for:
Designed by:
On:
Day
Month
Year
Iteration:
No.
Key Partners
Who are our Key Partners?
Who are our key suppliers?
Which Key Resources are we acquiring from partners?
Which Key Activities do partners perform?
Key Activities
What Key Activities do our Value Propositions require?
Our Distribution Channels?
Customer Relationships?
Revenue streams?
Key Resources
What Key Resources do our Value Propositions require?
Our Distribution Channels? Customer Relationships?
Revenue Streams?
Value Propositions
What value do we deliver to the customer?
Which one of our customer's problems are we helping to solve?
What bundles of products and services are we offering to each Customer Segment?
Which customer needs are we satisfying?
Customer Relationships
What type of relationship does each of our Customer Segments expect us to establish and maintain with them?
Which ones have we established?
How are they integrated with the rest of our business model?
How costly are they?
Channels
Through which Channels do our Customer Segments want to be reached?
How are we reaching them now?
How are our Channels integrated?
Which ones work best?
Which ones are most cost-efficient?
How are we integrating them with customer routines?
Customer Segments
For whom are we creating value?
Who are our most important customers?
Cost Structure
What are the most important costs inherent in our business model?
Which Key Resources are most expensive?
Which Key Activities are most expensive?
Revenue Streams
For what value are our customers really willing to pay?
For what do they currently pay?
How are they currently paying?
How would they prefer to pay?
How much does each Revenue Stream contribute to overall revenues?
www.businessmodelgeneration.com

Figure 3: The Business Model Canvas

3 Preliminary Market Analysis

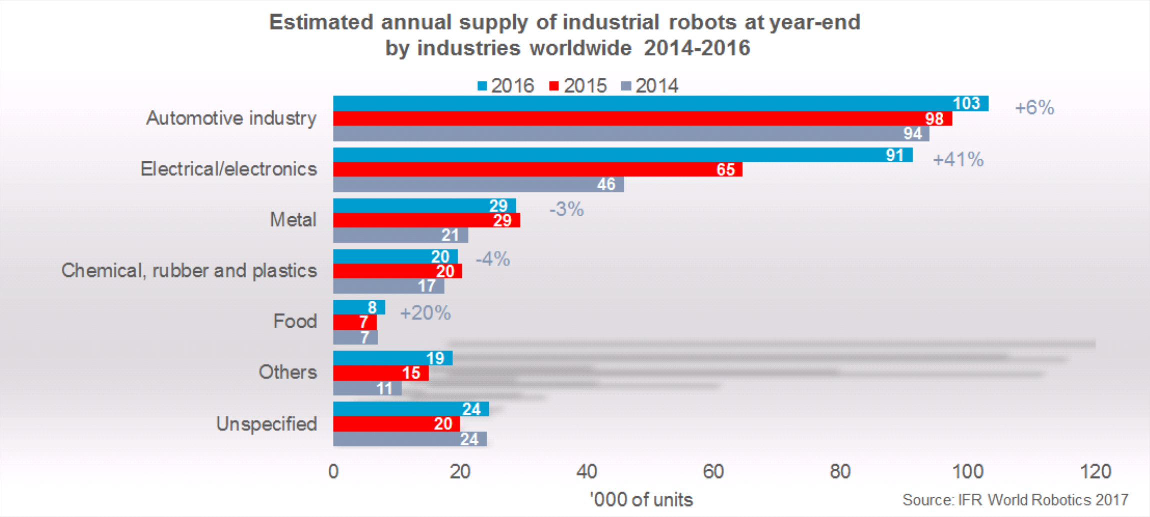
3.1 Market Size and Share

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.



Source: IFR World Robotics 2017

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.

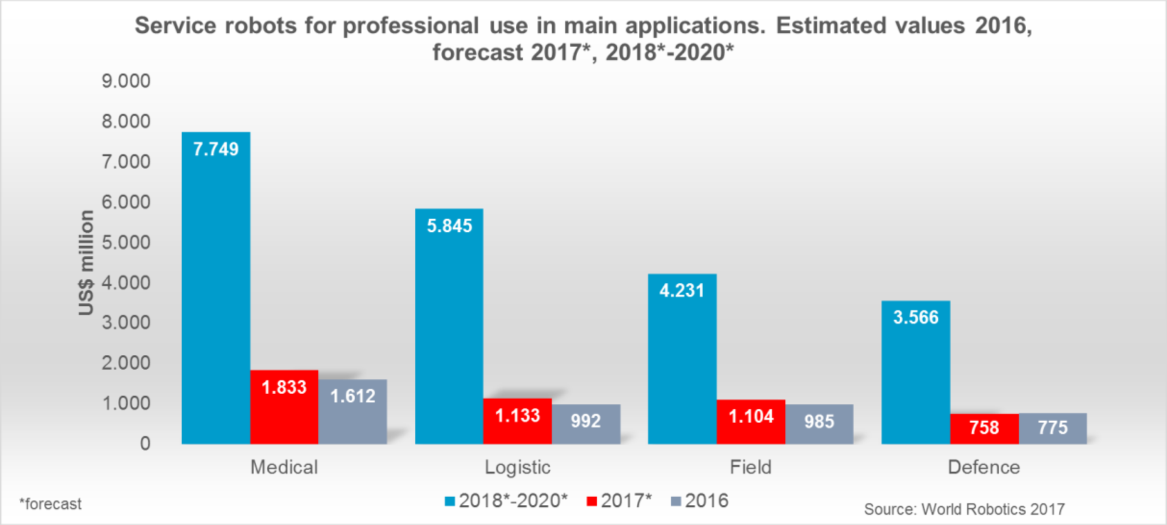


Figure 5: Main application sectors for professional service robotics [IFR, WorldRobotics 2017]

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.

3.3 Stakeholders and their Needs

The know-how 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 Stakeholders

Target Group	Examples of stakeholders
Industry: transport and logistics, manufacturing, healthcare, agriculture, and other possible domains that RobMoSys could have an impact.	OEMs, component suppliers, integrators of robotics systems, tool vendors, consulting and service providers, certification organizations, standardization groups and industrial forums. SMEs constitute a special interest group for RobMoSys, as they generally have very limited access to basic or applied research to develop new products.
Policy makers	Consultancy providers, assessor companies, standardization and regulation bodies.
Research community	Universities, research institutes.
Scientific communities	Robotic software development, quality assurance communities.
Open-source communities	Developers of open-source tools for CPS.
Users of Robotics Systems	People that will ultimately use robotics systems or their services.

RobMoSys must create value for them by helping them overcome the constraints (e.g., costs, resources, maintenance, reuse).

Table 2 provides a summary of needs per stakeholder type.

Table 2. Needs of RobMoSys Stakeholders

Stakeholders	Interested in:
Original Equipment Manufacturers (OEMs)	• Complying with the standards for development, assurance and certification. • 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.
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, and that ensure the integrity of the non-functional properties passed through the supply chain. • Tools that support these processes.
Consulting and Service Providers	• RobMoSys results that 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	• Assurance and certification process.
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.

3.4 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.

3.4.1 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.

3.4.2 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.

3.4.3 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.

3.4.4 Difficult integration

The RobMoSys platform might result 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.

3.4.5 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.

3.5 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 Preparation of RobMoSys Business Models

The goal of this section, as a second iteration of RobMoSys business models, is **identifying the RobMoSys user profiles and the value propositions for them**. We plan to use this information to discuss with relevant RobMoSys stakeholders and experts during the last period of the project.

4.1 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.

4.1.1 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 6.

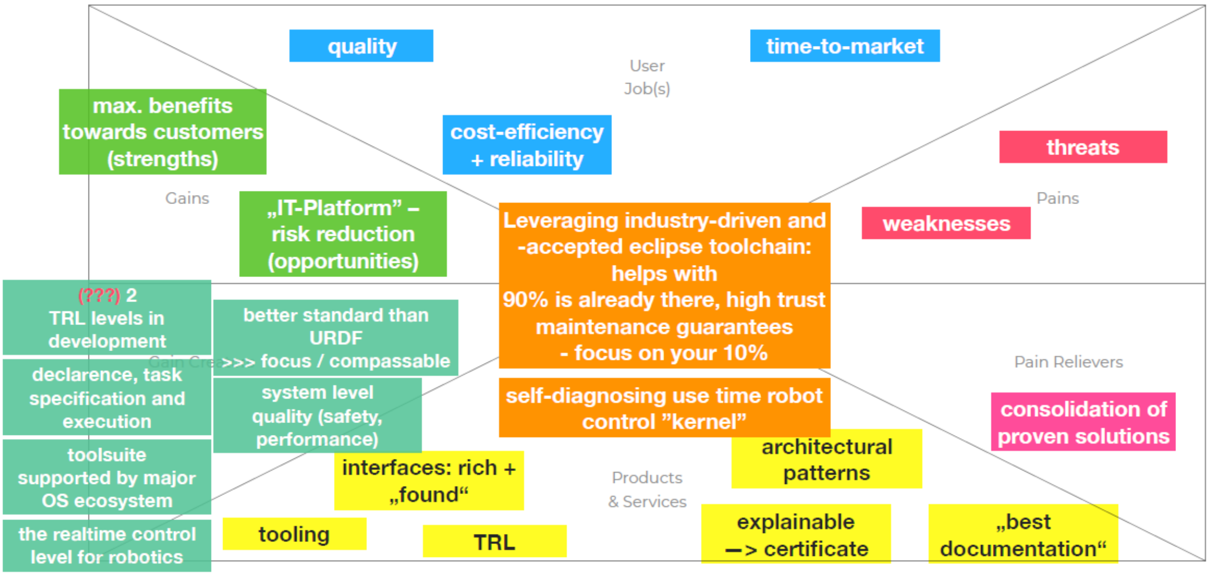

quality
time-to-market
User
Job(s)
max. benefits towards customers (strengths)
cost-efficiency + reliability
threats
Gains
„IT-Platform" – risk reduction (opportunities)
Leveraging industry-driven and -accepted eclipse toolchain: helps with
90% is already there, high trust maintenance guarantees
- focus on your 10%
weaknesses
Pains
(???) 2 TRL levels in development
better standard than URDF >>> focus / compassable
Gain Crea
Pain Relievers
declarence, task specification and execution
system level quality (safety, performance)
self-diagnosing use time robot control "kernel"
consolidation of proven solutions
toolsuite supported by major OS ecosystem
interfaces: rich + „found"
architectural patterns
Products & Services
the realtime control level for robotics
tooling
TRL
explainable —> certificate
„best documentation"

Figure 6: Value Map for Component Suppliers

4.1.2 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.

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 7.

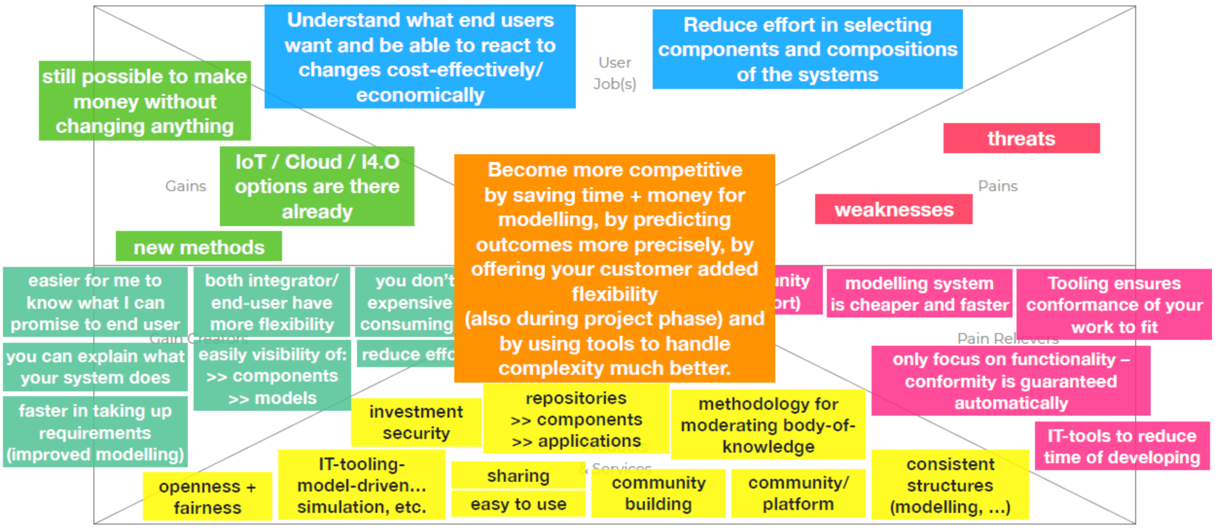

Understand what end users want and be able to react to changes cost-effectively/ economically
User Job(s)
Reduce effort in selecting components and compositions of the systems
still possible to make money without changing anything
Gains
IoT / Cloud / I4.O options are there already
new methods
threats
Pains
weaknesses
Become more competitive by saving time + money for modelling, by predicting outcomes more precisely, by offering your customer added flexibility (also during project phase) and by using tools to handle complexity much better.
easier for me to know what I can promise to end user
both integrator/ end-user have more flexibility
you don't expensive consuming
Gain Creators
you can explain what your system does
easily visibility of: >> components >> models
reduce effo
faster in taking up requirements (improved modelling)
investment security
repositories >> components >> applications
methodology for moderating body-of-knowledge
openness + fairness
IT-tooling- model-driven... simulation, etc.
sharing
easy to use
Products & Services
community building
community/ platform
unity ort)
modelling system is cheaper and faster
Tooling ensures conformance of your work to fit
Pain Relievers
only focus on functionality – conformity is guaranteed automatically
IT-tools to reduce time of developing
consistent structures (modelling, ...)

Figure 7: Value Map for System Integrators

4.1.3 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 8.

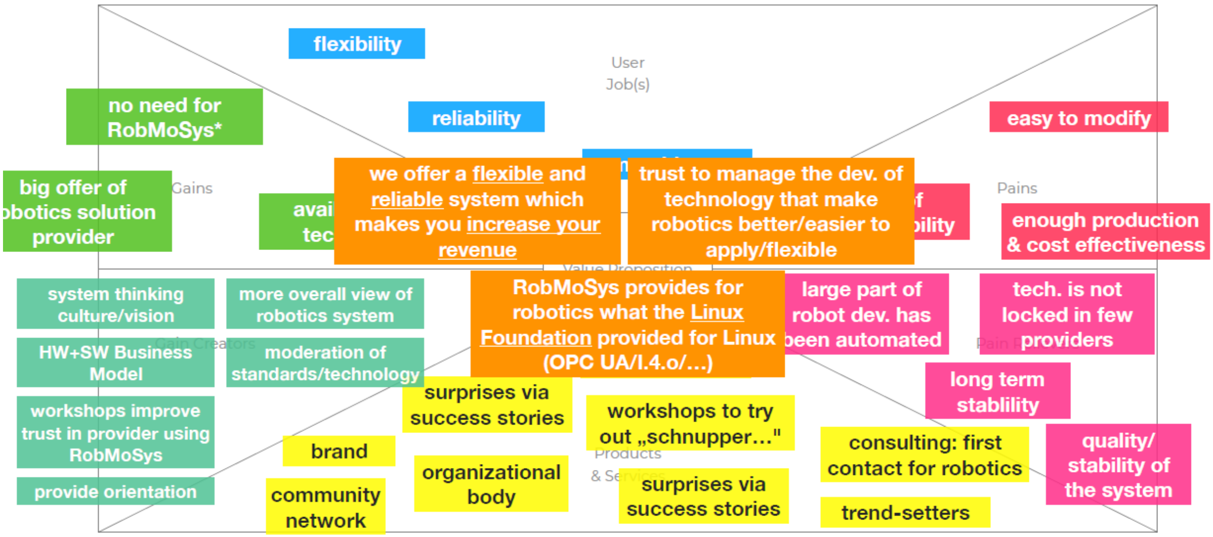

flexibility
User
Job(s)
no need for RobMoSys*
reliability
easy to modify
big offer of obotics solution provider
Gains
avai
tec
we offer a flexible and reliable system which makes you increase your revenue
trust to manage the dev. of technology that make robotics better/easier to apply/flexible
f
bility
Pains
enough production & cost effectiveness
Value Proposition
system thinking culture/vision
more overall view of robotics system
RobMoSys provides for robotics what the Linux Foundation provided for Linux (OPC UA/I.4.o/…)
large part of robot dev. has been automated
tech. is not locked in few providers
HW+SW Business Model
Gain Creators
moderation of standards/technology
Pain Relievers
long term stablility
workshops improve trust in provider using RobMoSys
surprises via success stories
workshops to try out „schnupper…"
brand
consulting: first contact for robotics
quality/ stability of the system
Products
& Services
organizational body
provide orientation
community network
surprises via success stories
trend-setters

Figure 8: 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.

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

4.2 RobMoSys Adoption Path

RobMoSys defines a process for stepwise increased adoption levels of the RobMoSys approach and community interaction (Figure 1).

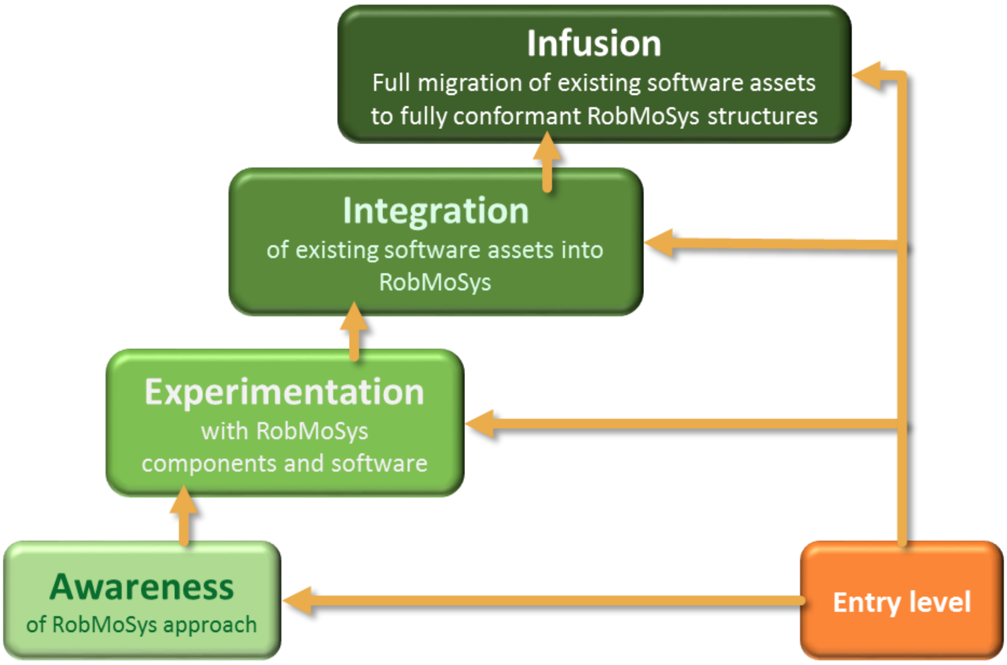

Infusion
Full migration of existing software assets to fully conformant RobMoSys structures
Integration
of existing software assets into RobMoSys
Experimentation
with RobMoSys components and software
Awareness
of RobMoSys approach
Entry level

Figure 1. 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.

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.