# ROS-Industrial Consortium Roadmapping

## Summary

To do

## Purpose

This document summarizes the ROS-Industrial Consortium (RIC) Americas efforts to provide a technical roadmap for the ROS-Industrial open source project. It consolidates the input of nearly two dozen member organizations that met in person and virtually during 2014. The roadmap attempts to identify technical needs based on end-user requirements for automation and robotics. Given these needs, the capabilities of the ROS-Industrial project can be mapped and gaps identified. Ultimately, this roadmap can then be used to allocate resources to address these gaps.



RIC Membership as of July 2014

## Scope

The scope has been limited to potential ROS-Industrial end-user needs, as expressed by the consortium members, and does not directly address developer, integrator or other stakeholders needs. It is prioritized based on feedback from current ROS-Industrial Consortium members, but may be revised as additional members join the consortium and additional user needs are identified. It is envisioned as a living document that requires periodic updates. The process is intentionally limited to addressing technical needs, although some non-technical requirements such as training and documentation have been identified. Specific implementation or development plans are also avoided since these may be dependent on funding sources and participating organizations.

## Process

The process roughly follows the [Sandia National Lab Fundamentals of Roadmapping](SandiaFundamentalsOfRoadmapping.pdf) technique. In summary, the steps include:

1. Define the scope and participants
2. Create a common vision for the product/technology
3. Identify stakeholder requirements
4. Define technology areas
5. Identify alternatives and gaps
6. Recommend path(s) forward
7. Evaluate roadmap
8. Develop implementation plans

### 1. **Define Scope and Participants**

See above for scope and RIC member participants

### 2. Create a Common Vision for the Product/Technology

This document does not attempt to take ownership of the ROS-Industrial Project vision; instead one might look towards the [project website](http://rosindustrial.org), or more broadly, [ROS.org](http://ros.org). The following is offered to gain alignment and a common language:

**Vision:** ROS-Industrial provides an open and flexible framework for advanced robotics development that:

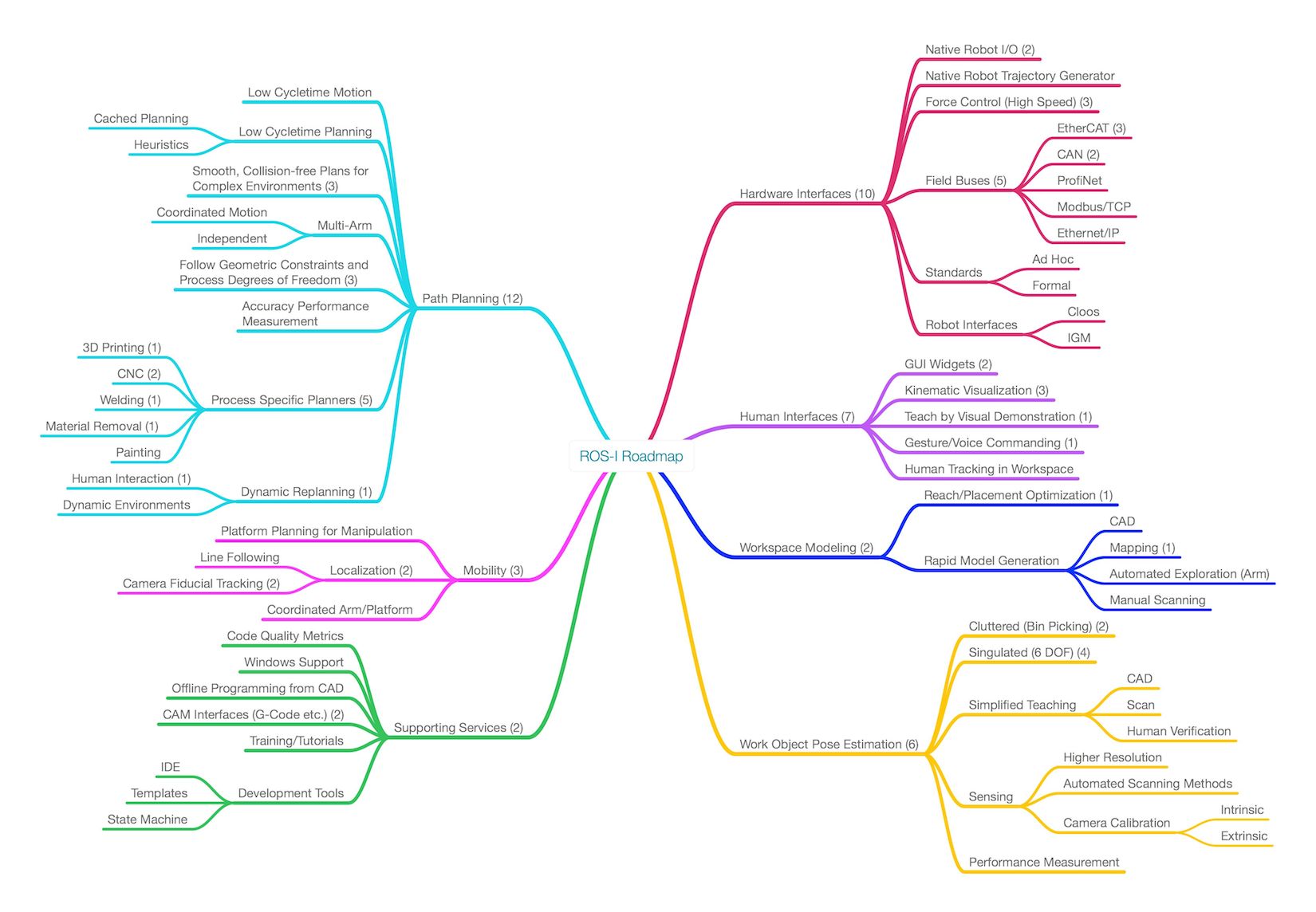
* Enables cross-platform compatibility
* Creates new applications that were previously infeasible or impractical
* Advances manufacturing productivity
* Improves worker well being
* Motivates students and researchers to focus on industrial problems
* Reduces implementation costs
* Foster growth of commercial developments

### 3. Identify Stakeholder requirements

This was accomplished through a use case elicitation process, which is summared in the [Use Cases](UseCases.md). Ten broad use cases were identified.

### 4. Define Technology areas

The technology areas were extracted from the [Use Cases](https://github.com/ros-industrial-consortium/roadmapping/blob/master/UseCases.md) and then defined through a series of [virtual brainstorming meetings](https://github.com/ros-industrial-consortium/roadmapping/tree/master/Meetings). The result is summarized in the mind map below.

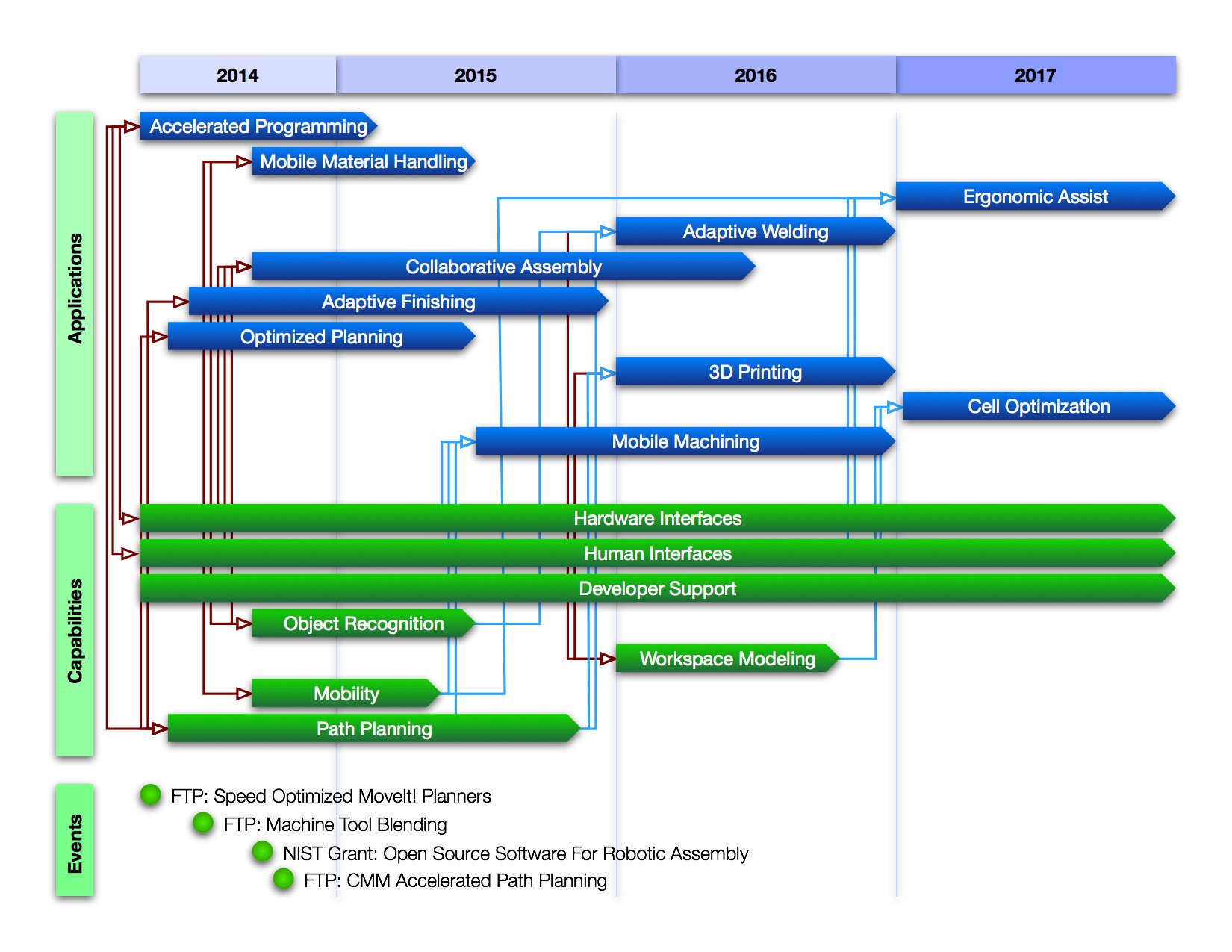


ROS-Industrial Technology Area Summary

One can observe that the areas were categorized into six technical subtopics and one non-technical topic (Supporting Services). The numbers in parethesis represent the number of use-cases that have needs in these areas. The topic areas are:

1. **Hardware Interfaces:** new interfaces to other robot contorllers, sensors and actuators
2. **Human Interfaces:** new ways to interact with humans by "traditional" means such as graphical interfaces or non-traditional means such as gesture commanads
3. **Workspace Modeling:** rapid generation of environement models and how to configure the robot within those environments
4. **Work-Object Pose Estimation:** identification and locating known objects in unknown locations (unstructured)
5. **Path Planning:** determining optimal motion plans for a robot manipulator while obeying process constraints
6. **Mobility:** topics unique to mobile platforms including localization and navigation
7. **Support Services:** developer and user support such as training and documentation

The technology areas (capabilities) mapped against the use cases (applications) and schedules are show in the graphic below:



ROS-Industrial Roadmap

The events shown have direct impact on the application areas and will result in development within many of the capabilities areas. The red connector lines represent concurrent development activities between projects and the capabilities. The blue lines represent dependencies between technologies and their application.

### 5. Identify Alternatives and Gaps

The summaries above provide a high level overview of the needs and potential technology solutions. In reality, each application will have detailed requirements that need to be mapped against features and performance of various tools within the ROS-Industrial community. This section attempts to define more specifically some of those needs.

#### Path Planning

The path planning subtopic is a cross-cutting and high priority need for the ROS-Industrial community. Although the broader ROS community continues to focus significant efforts in this area, many of the problems that are of interest to researchers and "field" roboticists involve operation in highly unstructured domains. These environments are characterized by clutter, uncertainty, and lack of *a priori* knowledge. Most industrial applications are afforded more structure such as conveyors, fixturing, and static workcells. Thus, industrial users expect higher performance as measured by cycle time, reliability, and determinism, at the expense of ability to accomodate greater uncertainty in the environment.

##### Alternatives

There are many competing alternatives for articulated arm path planning that include:

* [MoveIt!](http://moveit.ros.org): The MoveIt! framework is actually a higher level construct that provides the facilities for various path planning codes.
* [OPML](http://ompl.kavrakilab.org): OMPL is library of sampling-based planners that are the defaults for ROS.
* [SBPL](http://wiki.ros.org/sbpl): The search-based planning library is a graph search framework.
* [OpenRave](http://openrave.org): OpenRave is a full environment for planning and simulation including a plug-in architecture for other codes.
* [CHOMP](http://wiki.ros.org/chomp_motion_planner): Optimization based planners.
* [Orocos/KDL](http://www.orocos.org/kdl): Orocos is a larger framework for robot control including state estimation and realtime components. KLDL provides kinematics solvers including trajectory planners.

Most of these alternatives are actually frameworks or libraries that permit extension and adaptation for varied planning algorithms. As such, all could be employed for ROS-Industrial applications. However, MoveIt! is the leading framework within the ROS community moving forward. Consequently, it is the recommended development environment.

##### Gaps

Among these frameworks, numerous planning algorithms are possible, but most are inappropriate for industrial problems. For example, the [RRT](http://en.wikipedia.org/wiki/Rapidly_exploring_random_tree) (Rapidly-Exploring Random Tree) algorithm is a common sampling-based planner that efficiently searches through high dimensional configuration spaces to find planning solutions. It can do so while handling geometric constraints. However, it fundamentally searches for a single goal state from an initial state. This may be acceptable for some material handling applications where only one pick and place location is needed, but many industrial processes have requirements for following an entire Cartesian path with known accuracy. An example is a welding application where the robot is required to follow a known position trajectory, perhaps with some flexibility in the weld gun orientation. In general, industrial path planners have the following desirable features:

* Able to follow arbitrary Cartesian paths with known accuracies (not just end goals)
* Able to permit flexible orientation for some orientation degrees of freedom, generally within process limits
* Able to constrain velocities along the Cartesian path
* Able to plan collision free trajectories along the entire path

None of the known path planners directly provide this set of capabilities and this is a significant barrier for implementing ROS-Industrial applications for processes such as painting, welding, material removal, deposition and other path-based plans. Note that there is ongoing work within the ROS-Industrial Community to address some of these needs including an [optimizing inverse kinematics trajectory planner](http://wiki.ros.org/industrial_moveit) and a [hybrid Cartesian planner](https://github.com/ros-industrial-consortium/descartes).

In addition, several other needs were identified such as multi arm coordinated planning and cycle-time optimized planning. Both of these are addressed in some capacity [here](https://github.com/ros-industrial/motoman/tree/hydro-devel/motoman_sda10f_support) for dual arm Motoman support and [here](http://static.squarespace.com/static/51df34b1e4b08840dcfd2841/t/53c3435ce4b00a24507b341d/1405305692771/IDEXX_Planning_Presentation.pdf) for cycle-time optimized planning. More work is require to broaden and generalize these solutions.

In summary, path planning is cross-cutting need (nearly all the use cases have driving requirements) and there is significant opportunity to make a contributions to these goals. This need is a high priority for ROS-industrial users.

### 6. Recommendations and Priorities

To Do