# Title : PolarSys technology vision becomes reality with Luna !

## A vision for extensible tool chains for Embedded Systems

In 2011, even before starting PolarSys officially, the team[[1]](#footnote-1) designed the first version of our technology vision:



It is the fundamental vision behind PolarSys development and collaboration. What you should remember is that:

* PolarSys relies on Eclipse in terms of its infrastructure and platform.
* PolarSys provides and catalogs technology bricks that provide single features like modeling support, documentation and code generation and model verification.
* PolarSys leverages these technology bricks into engineering tools that fully address one specific activity of Embedded System Engineering e.g., Architecture Design, Development or Testing
* Tools vendors, service providers or large organizations (mainly their methods and tools teams), put together engineering tools and add their specific features to add extra value to the tool chain, or to adapt the tools to a specific company workflow and environment.

In 2014, this vision becomes reality as major PolarSys projects are about to publish their 1.0 version and reach hence a new level of maturity.

During last EclipseCon in USA, we were doing a status about our technologies and designed the following image:



The diagram shown on the previous figure is not about the actual size of each component, but it puts forward the fact that the shared technologies between the different tools (the Eclipse platform, Eclipse modeling technologies and the shared components) represent a significant piece of software and millions of line of code that we leverage to build the tools.

First, this diagram exactly shows the different kind of tools we are creating for the Polarsys engineering layer, among them:

* Papyrus for UML/SysML modeling,
* CDT for C++ developing,
* Polychrony is an open-source synchronous modeling environment
* Capella, a system engineering tool that wi**l**l be contributed in the coming months by the Thales, …

What’s cool is that, with Luna, we are in the unique situation to deliver 1.0 version of two major kinds of Modeling projects:

* Papyrus, the Eclipse project for UML and SysML reaches its awaited 1.0 version. In this context, he project provides advanced customization features on top of UML and SysML. The team is now focusing on working on better user experience, and more documentation that will be ready for Luna (More details in this specific article about Papyrus).
* Sirius reaches its 1.0 version and makes it much easier and quicker than GMF to develop specific modeling tools (See the Sirius newsletter). On top of that, Thales is open sourcing KitAlpha for the definition of multi-viewpoint oriented tools, and will open source Capella, the system engineering tool they use in several significant projects.
* Polychrony is being integrated as the forthcoming Eclipse project POP. It offers a synchronous modeling infrastructure to verify or transform models or generate C, Java code from e.g. AADL, Geneauto Simulink or project P models. It is currently being used to develop one of the MoCCs of the forthcoming AADL timing annex.

We also share more and more components like Acceleo for code generation, Gendoc for documentation generation, OCL for enabling model verification, Reqcycle for requirements engineering, (add the links to the projects or projects proposals) and the most interesting projects from the Topcased (link to Topcased) community finally migrated to PolarSys.

# Join us and contribute!

Of course, it is still a work in progress, and some of you are impatient to get be able to use these tools. I would say that it is certainly the right time to step in the community.

First, to challenge us with your expectations, and to submit bugs, request for documentation or to ask questions on the mailing list.

Second, we are starting to work an idea we have since several months: our [integrated demo](https://polarsys.org/integrated-demo).

The goal is to design, develop and build a nice and extensible rover system and to show how the PolarSys tools help design this system.

The process will be iterative and fully open. We will collect new ideas, and we will regularly select several ideas to inject in the next iteration.

1. Mainly Benoît Langlois from Thales and Alain Rossignol from Airbus Defense and Space [↑](#footnote-ref-1)