

Systems Engineering Standards and Open Source Tools for Very Small Enterprises – A Case Study

Claude Y. Laporte
École de technologie supérieure
Department of Software and IT Engineering
1100 Notre-Dame Street West, Montréal
Québec H3C 1K3, Canada,
Claude.Y.Laporte@etsmtl.ca

Ronald Houde
Mannarino Systems & Software
100 Alexis-Nihon Blvd, Suite 650
Saint-Laurent, Québec H4M 2P2, Canada
Ronald.Houde@mss.ca

Gaël Blondelle
Eclipse Foundation Europe
Annastrasse 46
D-64673, Zwingenberg, Germany
gael.blondelle@eclipse.org

Abstract. Very small entities (VSEs) play an increasingly important role in the global economy. The products they develop are often integrated into products made by larger enterprises. The “Big League” clients, furthermore, demand of the VSEs that they assume a much broader role, spanning the entire development life-cycle of the product instead of being limited to a “build-to-print” approach. To address this new reality, to exploit the lean and efficient nature of VSEs and to adapt to their typical budget and resource constraints, the ISO/IEC 29110 systems engineering standards, management and engineering guides were developed from ISO/IEC/IEEE 15288. In addition, and by design, the standard is supported by Deployment Packages, software tools and training kits. The INCOSE VSE working group developed the deployment packages. A deployment package is a set of artefacts designed to facilitate the implementation of a standard or a set of practices in a VSE. In tune with the need for low cost and flexibility, Open Source software tools are emerging in PolarSys to complete “Big League” development life-cycle toolsets, which are often out-of-reach to VSEs. Finally, to make the deployment of the standard possible in the VSE, training packages, supported by relevant pilot projects help VSE personnel learn how to apply all of the above.

Introduction

Industry recognizes the value of Very Small Entities (VSEs), i.e., enterprises, organizations, departments or projects with up to 25 people, in contributing valuable products and services. A large majority of enterprises worldwide are VSEs. In Europe, for instance, as illustrated in Table 1, over 92% of enterprises are micro-enterprises. They have fewer than nine employees. Micro enterprises account for 70% to 90% of enterprises in OECD countries and about 57% in USA.

Table 1: Size of enterprises in Europe (Moll 2013)

Type of enterprise	Number of employees	Annual turnover (EURO)	Number of enterprises (% of overall)	Number of enterprises
Micro-enterprises	1 - 9	≤ 2 million	92.2 %	19 968 000
Small enterprises	10 - 49	≤ 10 million	6.5 %	1 358 000
Medium enterprises	50 – 249	≤ 50 million	1.1 %	228 000
SMEs, total	87 100 000		99.8 %	21 544 000*
Large enterprises	> 250	> 50 million		
Large enterprises, Total	42 900 000		0.2 %	43 000
* Independent companies only, excluding legally independent companies that are part of large enterprises.				

Many international standards, such as ISO/IEC/IEEE 15288 (ISO 2015a), have been developed to capture proven management and engineering practices. In the last decades, under risk sharing partnerships meant to dilute the cost of developing complex products, large companies have progressively moved into risk sharing partnerships with their suppliers. As a result, “partners began to take over not only investments in tools and non routine engineering and infrastructure but also to participate more directly in investments and project development, thereby acquiring rights to their future sales income.” (Figueiro et al 2008). In that transfer of responsibilities, for example, specifications traditionally prepared by the upper tier client, now had to be developed by their subordinate lower tier risk sharing partners. In sectors such as aerospace, medical devices, automotive, rail transportation, atomic energy and industrial processes, the transfer entailed the production of a range of process artefacts that range from plans, requirements specification, design descriptions, source code/netlists and test procedures. Over time, such risk sharing partnerships were pushed down to lower tier suppliers in the supply chain, reaching progressively smaller businesses that were ill-equipped to deal with the complete span and life-cycle of product development activities integrated with higher tier processes and tools. The existing system and software development life-cycle standards, furthermore, were not written with VSEs in mind, were difficult and burdensome to tailor and apply in such settings and require staff resources to manage, thereby increasing the bureaucratic burden and overhead costs. A need therefore arose to provide VSEs with solutions that would keep bureaucratic and overhead costs down while achieving the performance and maturity benefits sought by larger organizations of existing international engineering standards, along with the concepts, processes and practices involved. An ISO Working Group (ISO/IEC JTC1 SC7¹ Working Group 24) was established to address these challenges by developing standards and guides adapted to the needs of VSEs.

This paper presents the application of the new systems engineering ISO/IEC 29110 standard (ISO 29110 hereon) developed for VSEs performing systems development with up to 25 people. More specifically, we will:

- Provide an overview the ISO 29110 standards and guides for VSEs;
- Describes the set of nine (9) Deployment Packages (DPs) covering the span of the systems engineering life-cycle and describe in more detail the Requirements Engineering (RE) DP;

¹ International Organization for Standardization/ International Electrotechnical Commission Joint Technical Committee 1/ Sub Committee 7

- Describe Open Source tools available to support both the RE Deployment Package, with the Eclipse Requirements Management Framework (RMF), and the Functional & Physical Architecture Deployment Package with Eclipse Papyrus for SysML;
- In the context of a Pilot Project, show how a collaborative Training Kit was developed to help VSEs implement the ISO 29110 Requirement Engineering Deployment Package.

Overview of ISO 29110 Systems Engineering Standards and Guides

The first version of ISO 29110 is a four-stage roadmap, also called profiles (Entry, Basic, Intermediate, Advanced) is applicable to VSEs that do not develop critical systems. VSEs targeted by the Entry profile are those working on small projects (e.g., at most six person-months of effort) and for start-ups. The Basic profile describes the development practices of a single application by a single project team with no special risk or situational factors. The Intermediate profile is targeted at VSEs developing more than one project in parallel with more than one team within the organization. The Advanced profile is targeted at VSEs wishing to sustain and grow as independent competitive system and/or software development business (ISO 2016).

The ISO 29110 documents are targeted by audience as described in Table 2.

Table 2: ISO 29110 target audience (ISO 2016)

ISO/IEC 29110	Title	Target audience
Part 1	Overview	VSEs and their customers, assessors, standards producers, tool vendors and methodology vendors.
Part 2	Framework and taxonomy	Profile producers, tool vendors and methodology vendors. Not intended for VSEs.
Part 3	Certification and Assessment guide	VSEs and their customers, assessors, accreditation bodies.
Part 4	Profile specifications	VSEs, customers, standards producers, tool vendors and methodology vendors.
Part 5	Management and engineering guide and service delivery guides	VSEs and their customers.
Part 6	Management and engineering guides not tied to a specific profile	VSEs and their customers.

Parts 1, 3 which are freely available from ISO, are the same for both the software engineering (SW) and systems engineering (SE) domains, while parts 4 and 5 have different content for these two domains. Parts 5, the management, engineering and service delivery guides are freely available from ISO². The future ISO 29110-6-x will provide management and engineering guides not tied to a specific profile.

Finally, it is worth mentioning that, even though the processes are described according to a sequential/waterfall approach, the ISO 29110 series is not intended to dictate the use of any particular life cycle such as waterfall, iterative, incremental, evolutionary or agile.

The ISO 29110 standard has a formal certification process. This process has been created taking into account the needs of software and systems development VSEs, so that audits should not be too expensive or time consuming. The certification scheme is based on ISO standards for certification body requirements and auditor capability requirements. A VSE is expected to invest about 85 hours for the certification process. The cost of an ISO 29110 initial audit is about \$1,500. This audit cost does not include the auditor's travel expenses. It is estimated that the cost of each one of the 2 surveillance audits, excluding the travel costs of the auditor, will be about \$1,200. The effort and expenses invested by a VSE for an audit is quite small compared to a typical CMMI-DEV assessment (Garcia 2015).

The reader will find more details about the development of ISO 29110 systems engineering standards and freely available guides in papers presented at the 2012 and 2015 INCOSE Symposia (Laporte et al. 2012, 2014a, 2015a).

Engineering and Management Guide for the Systems Engineering Basic Profile

The systems engineering Basic profile, illustrated in Figure 1, is composed of two processes: Project Management (PM) and System Definition and Realization (SR). As defined in ISO 29110, the purpose of the PM process is to establish the system engineering tasks and carry them out in a systematic way, which makes it easier to meet the project objectives with regard to expected quality, time and cost. The purpose of the System Definition and Realization (SR) process is the systematic performance of the analysis, design, construction, integration, verification and validation activities of new or modified systems according to the specified requirements.

As illustrated in figure 1, the PM process uses the acquirer's Statement of Work (SOW) to establish the project plan and develop the product requested using the SR process.

² Available at no cost from ISO: <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

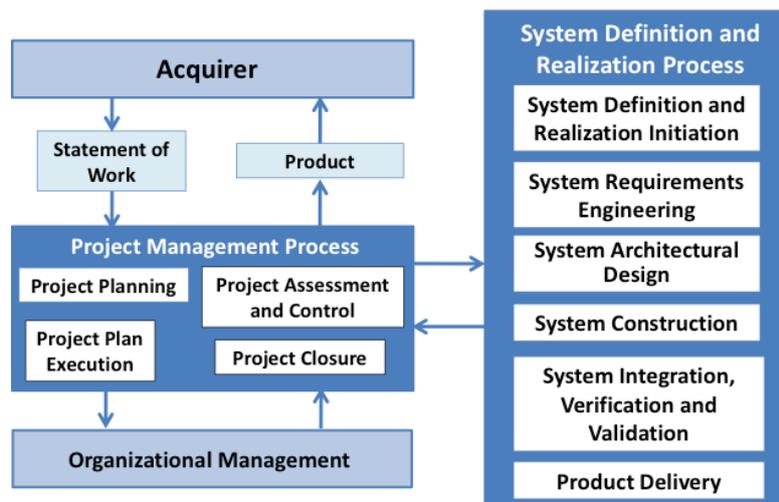


Figure 1. Processes of the systems engineering Basic Profile (Laporte 2014b)

The ISO 29110 standards and guides for systems engineering are designed to work hand-in-hand with the ones for software engineering.

Deployment Packages to Support the Systems Engineering Basic Profile

The INCOSE VSE Working Group defined a set of guidelines explaining in more detail the processes defined in the Basic profile. These guidelines are freely accessible to VSEs on the Internet as a collection of Deployment Packages (DPs). A DP is a set of artefacts developed to facilitate the implementation of a set of practices for the selected framework in a VSE (Laporte 2014b). Since the INCOSE handbook is a 'how to' document, it was used to develop the set of DPs. DPs are designed such that a VSE can implement its content without having to implement the complete Basic profile at the same time. The table of contents of a Deployment Package is illustrated in Table 3.

Table 3: Table of contents, SE Deployment Package (adapted from Laporte et al. 2012)

1. Introduction
Purpose of this document
Key Definitions
2. Why this process is important
3. Overview of main tasks
3.1 Tasks
3.2 Roles and artefacts
3.3 Activity life cycle and examples of life cycles
Appendix A Templates
Appendix B Checklists
Appendix C Coverage Matrices (ISO 15288, ISO 9001, CMMI)
Appendix D Tools
Appendix E References
Appendix F Evaluation Form

Figure 2 illustrates the set of SE DPs for the Basic profile, which is available on the Internet³ and on the INCOSE VSE page.

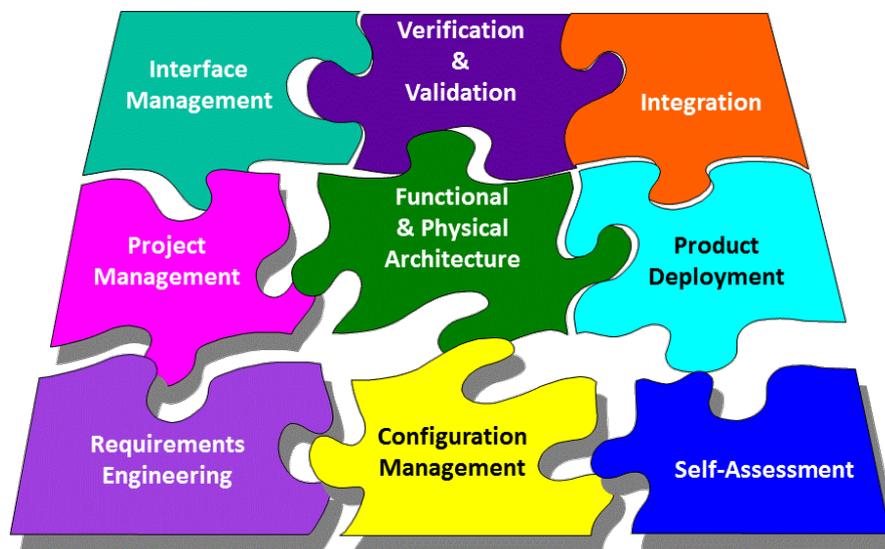


Figure 2. Deployment Packages for the SE Basic profile

The *Requirements Engineering DP* defines a collection of activities, tasks, steps, roles, artefacts, templates and checklists for those work products that achieve the objectives of ISO 29110 for:

- the definition of Acquirer/Stakeholder Requirements;
- the definition of System Requirements;
- the decomposition of System Requirements into System Element Requirements using the architecture defined in the Functional & Physical Architecture DP;
- the creation of traceability links between each level of requirement decomposition;
- the preparation of the System Specification;
- the preparation of System Element Specifications (for hardware or software sub-systems); and
- the verification and validation of the requirements.

The Requirements Engineering DP identifies candidate Requirements Management Tools suitable for VSEs and provides instructions for tailoring the tools to support the DP. One of those tools is the Eclipse Requirements Management Framework (RMF). Two checklists are provided to evaluate the quality of the requirements developed by the VSE and the specification assembling and structuring the collection of individual requirements. The checklists apply world-class standards such as ISO/IEC/IEEE 29148 (ISO 2011) and the INCOSE Guide for Writing Requirements.

The *Functional & Physical Architecture DP* achieves the same purpose as the Requirements Engineering DP for:

- the conduct of system architecture definition trade studies (functional architecture);
- the definition of System Elements and the interfaces between them (physical architecture);
- the verification of the System Architecture and Sub-System Specifications.

³ <http://profs.etsmtl.ca/claporte/English/VSE/index.html> (Deployment Packages)

The *Interface Management* DP repeats the process for the definition, management and control of interfaces at all levels of the system.

The *Integration DP* shows how System Elements are assembled and integrated into a complete system in accordance with an Integration Plan.

The *Verification & Validation* DP details how a requirements-driven verification planning and execution process can be implemented by a VSE for a System and how that System is validated ultimately against captured Acquirer/Stakeholder Requirements.

The *Product Deployment* DP defines how the System is produced and deployed with the Acquirer, including the transition from an existing system to the new one and the training of users.

The *Project Management* DP shows how all the activities are planned and executed and risks (project or technical) are identified and managed according to the threat they represent to the project.

The *Configuration Management* DP defines how all the artefacts created by the VSE, including the System Elements and System itself are placed under configuration control.

Finally, the *Self-Assessment* DP provides instructions, templates and checklists allowing a VSE to evaluate how well they have applied the collection of DPs to their processes to achieve the objectives of the ISO 29110 Basic profile.

Open Source Tools Developed to support DPs

One of the primary constraints of VSEs is the limited access their resources and staffing permit them in terms of life-cycle development tools. Most cannot afford the burden of “Big League” tools and have been relegated to using word processing and spreadsheet tools to manage requirements. With PolarSys⁴, a Working Group of the Eclipse Foundation created by large industry players and by tools providers to collaborate on the creation and support of Open Source tools for the development of embedded systems, VSEs can now have access to open source tools that support the various deployment packages defined in the ISO 29110 Basic profile. We detail here how VSEs can use two PolarSys open source solutions for Requirement Management and Functional & Physical Architecture.

Open Source tool for the Requirement Management Deployment Package

Current commercially available requirements management tools add two further constraints that will often discourage VSEs: 1) the tool’s flexibility requires of the RM tool that it be custom-fitted to the VSEs needs and that support staff or contractors be on hand to provide training, tailoring and implementation support throughout the span of the project; or 2) the RM tool implements a specific product development life-cycle process that is not compatible with the often “lean” processes used by the VSE. Whereas basic word-processing and spreadsheet tools can be acceptable for a small number of requirements and limited linking among them, their capabilities are rapidly challenged on medium to large size projects. After a somewhat shaky start, Open Source requirements management tools have taken a significant turn with the emergence of the Eclipse Requirements Management Framework

⁴ <http://www.polarsys.org>

(RMF)^{5,6}, the Object Management Group (OMG) Requirements Interchange Format (ReqIF)⁷ and the support of such working groups as PolarSys. However, those tools are often generic in nature and require some tailoring which is addressed in the RE Deployment Package.

RMF is a framework for working with textual requirements, structured as ReqIF models. RMF uses natively ReqIF, allowing you to exchange requirements with many industry applications like IBM Rational DOORS® or PTC integrity®. Using ReqIF as its native data storage structure provides another significant advantage for VSEs by allowing them to exchange ReqIF export files directly with upper tier partners and skipping the tool tailoring/customization step entirely. ProR was developed as a RMF Graphical User Interface (GUI) that allows manipulating requirements within a RMF environment. Operating within the Eclipse Integrated Development Environment (IDE), it allows a requirements engineer to open ReqIF files with a click, allowing you to immediately inspect and edit files. Powerful extensions exist for dealing with rich text, ReqIFz-Archives and others. Those extensions are made available in integration with UML/SysML modeling tools within PolarSys, illustrated in Figure 3, and openETCS⁸.

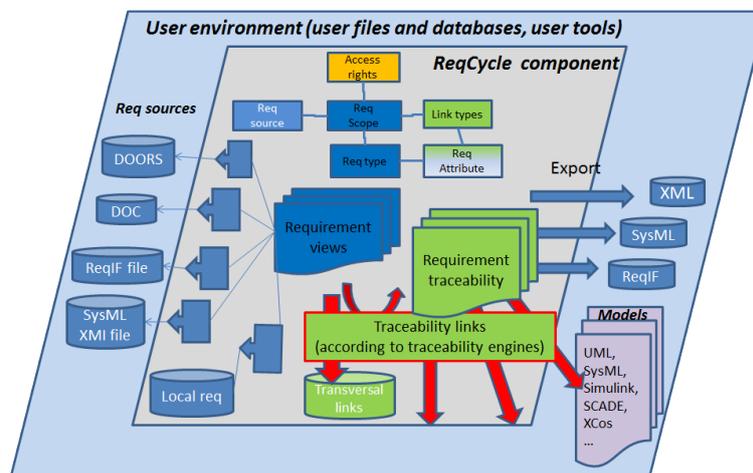


Figure 3. Polarsys ReqCycle Requirements Management Component

Open Source tools for the Functional and Physical Architecture Deployment Package

PolarSys offers two tools for Architecture design that can be used in the context of ISO 29110 Deployment Packages: Capella⁹ and Eclipse Papyrus¹⁰.

Capella is the tool developed by Thales to support Arcadia¹¹, a Model-Based Engineering Method for System, Software and Hardware Architectural Design. It has been open sourced in 2014 and recently reached its 1.0 version. Capella is largely deployed inside Thales for the engineering of complex systems in many domains including avionics, transport and space.

⁵ <http://www.eclipse.org/rmf/>

⁶ Under the leadership of Michael Jastram

⁷ <http://www.omg.org/spec/ReqIF/>

⁸ <http://openetcs.org/tool/>

⁹ <http://polarsys.org/capella/>

¹⁰ <https://eclipse.org/papyrus/>

¹¹ <http://polarsys.org/capella/arcadia.html>

We focus on the usage of Eclipse Papyrus, the Eclipse project that provides UML and SysML modeling. Papyrus reached its 1.0 version in June 2014 with the Eclipse Luna release train. In the last 5 years, Papyrus has benefited from a strong support from large organisations to improve its maturity, customizability and usability in different use cases, from small teams to larger projects.

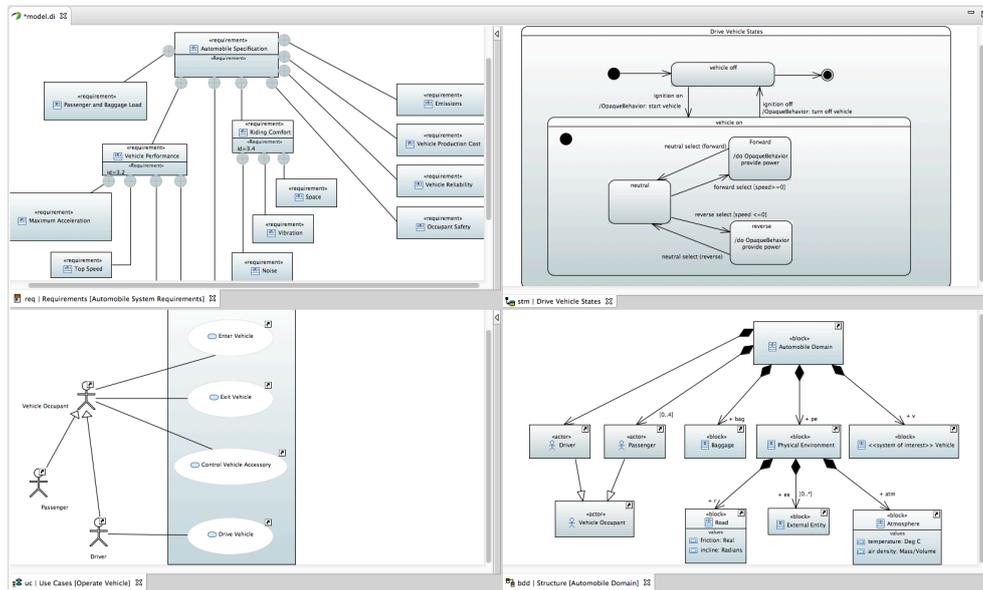


Figure 4. Eclipse Papyrus for SysML and UML

Training Kit and Pilot Project

At this stage, two of the three pieces of the equation are in place: the ISO 29110 and the Requirements Engineering Deployment Package provide the *Process* piece; an Open Source Framework, constructed around the Eclipse Requirements Management Framework, provides the *Tool* piece. We now need to bring *People* into the equation. A survey of enterprises done in (Land 1997) identified very specific needs they have in order to achieve an acceptance of standards. Those needs include:

- User training course
- Examples of deliverables
- Deliverable templates
- CASE tool support for documentation generation
- On-line or phone support
- Educators resource/support

An inexpensive, publicly licensed (i.e. allowing a VSE to tailor and adapt to its needs with a few constraints as possible) and easily deployable training package was therefore needed to allow VSEs to deploy an effective Systems Engineering life-cycle process. Such a Training Kit was developed with the Eclipse foundation¹², by a team of practitioners from around the world.

Since requirements are the cornerstone of a requirement-centric development life-cycle, it should not come as a surprise that the RE DP was the first of the Systems Engineering DP

¹² <http://jastram.github.io/teaching/>

selected for development. It will serve as the proving ground upon which the other Systems Engineering DPs will be developed under the “Systems Engineering for VSEs” INCOSE Working Group.

As a pilot project to use the DPs and tools, the RE DP Training Kit, has been constructed around Case Studies that can be extended to apply to all the Systems Engineering DPs and the entire life-cycle of a Systems Engineering effort (i.e. hardware, software, organizational processes, etc.). The Case Studies, used in the Training Kit, are: 1) the Traffic Light Control System¹³; and 2) the Autonomous Rover¹⁴.

The Autonomous Rover case was selected for a pilot project because it represents a typical System development problem (i.e. includes both hardware and software elements). Also, it is sufficiently simple that a complete solution can be developed during the training period and allows students to become proficient with the application of the System Requirements Engineering Process for a VSE, the artefact templates and the RE Tool.

The RE DP Training Material for either case has for objective to satisfy Goal 1 of Step 0 of the Autonomous Rover project, which is to gather the first set of requirements. The set, of which a sample is shown in Table 4, consists of functional, non-functional, hardware and safety requirements to be implemented in three (3) phases of the Autonomous Rover development project. Whereas the RE DP focuses on the management and engineering of textual requirements, the Case Study is designed from its inception to be extendable to Model-Base techniques, methodologies and tools.

Table 4: Sample of Autonomous Rover project requirements

Identifier	Description	Target Version	Category	Comment
ROVER_FUNC_010	The Rover must support various payloads (sensors/camera/robotic arm) thanks to a pluggable software architecture	1.0	Functional	
ROVER_HARD_010	The Rover must be built with the Polulu Dagu Rover 5 platform	1.0	Hardware	
ROVER_NFUNC_010	Battery must provide an autonomy of at least 10mn	1.0	Non Functional	Will evolve to 30mn autonomy in V3.0
ROVER_SAFE_010	The Rover must avoid crash in obstacles	2.0	Safety	

The Rover will be built with a low cost Polulu Dagu Rover 5 platform¹⁵ illustrated in Figure 5. The chassis includes a battery holder and two DC motors, with an independent drive train and a quadrature encoder for each tread. The outer dimensions of the chassis are approximately 24cm (9.5") long, 23 cm (9") wide, and 8 cm (3") tall in its default, flattened configuration.

¹³ <https://github.com/jastram/teaching/tree/master/SE>

¹⁴ https://polarsys.org/wiki/PolarSys_Rover_Demo

¹⁵ <https://www.pololu.com/product/1551>



Figure 5. The Autonomous Rover

Model-Based Requirements Engineering

In support of the ISO 29110 System Requirements Engineering DP, Use Case Analysis can be used to ensure system requirements properly fulfil user needs expressed in a document such as the Operational Concept (OpsCon) document. Figure 6 shows a Use Case Diagram that has been used to identify the main operational Use Cases for the Autonomous Rover. Those Use Cases were identified using the Autonomous Rover OpsCon.

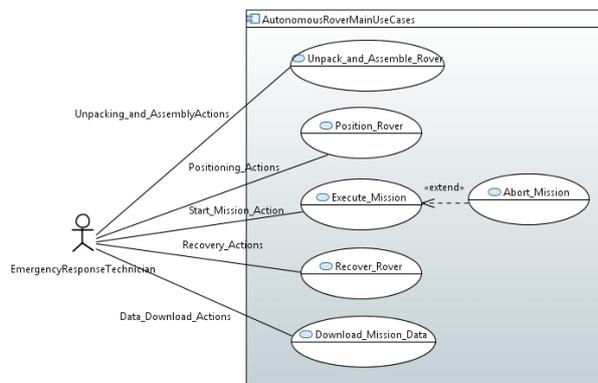


Figure 6 - Autonomous Rover Main Use Case Diagram

In Figure 7, we show how each Use Case Diagram is traced to the System Requirements it is meant to validate. In this case, a Requirements Diagram is used to show which System Requirements are validated by the “Execute_Mission” Use Case.

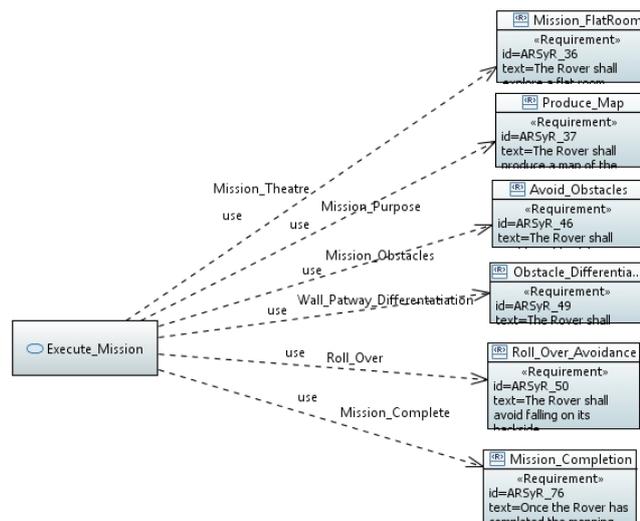


Figure 7 - Use Case to System Requirements Linking Requirements Diagram

Once this is accomplished, the Systems Engineer can proceed to the definition of Scenarios and related Sequence Diagrams, which will serve to ensure system requirements are coherent, complete, unambiguous and un-conflicting.

Model-Based Functional and Physical Architecture Definition

In parallel with the system requirements analysis, the Systems Engineer will, under the ISO 29110 System Functional & Physical Architecture DP, carry-out System Architectural trade studies and will build a second SysML model, the System Design Model. This model will comprised Block Definition Diagrams (BDD), such as the one in Figure 8, depicting the top-level functional architecture of the Autonomous Rover.

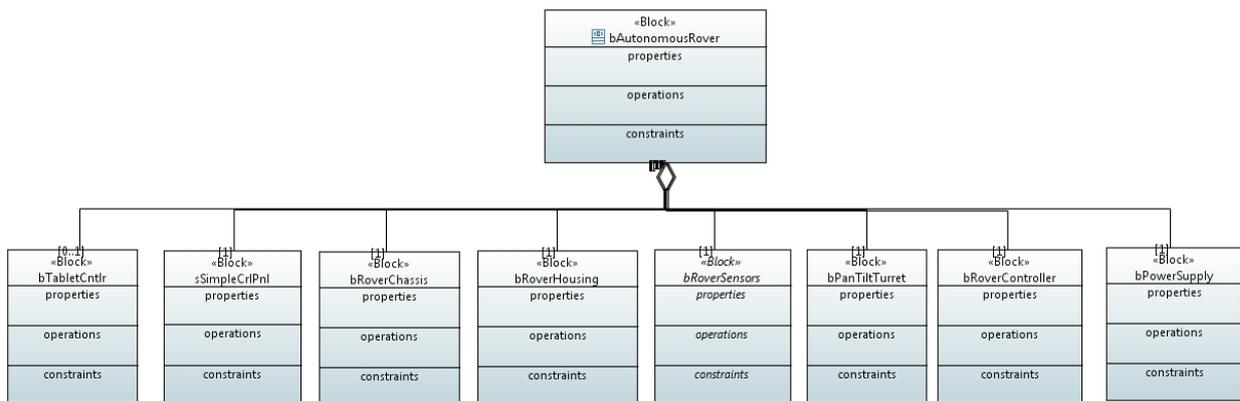


Figure 8 - Autonomous Rover Top BDD

Figure 9 shows a further level of detail, in this case for the Autonomous Rover Sensors.

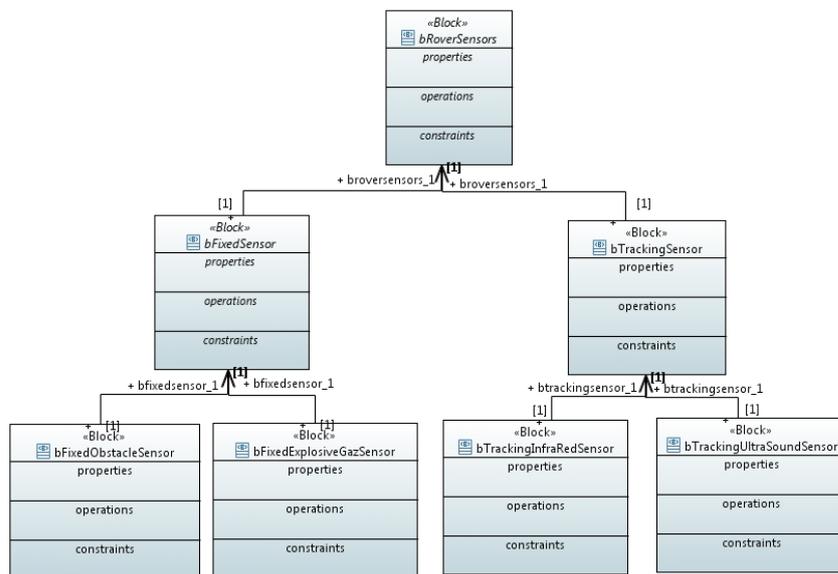


Figure 9 - Autonomous Rover Sensors BDD

The Internal Block Diagram (IBD) will be the final System Design Model Diagram that will capture the physical construction of the Autonomous Rover and the interfaces between the various sub-systems and components.

The training material has been collected in a GitHub project¹⁶ and is available under the Apache 2.0 License, which allows VSEs to use the material as is, or tailor it to meet their own needs. We also expect educators and trainers will use and improve the training material over time.

Conclusion and Future Work

Industry recognizes the contribution of VSEs in terms of the valuable products and services they offer. A large majority of organizations worldwide have fewer than 25 people. Most system and software engineering standards were not easily applied in VSEs, where they were generally found difficult to understand and implement.

An ISO working group has developed a set of standards and guides to address the needs of VSEs developing system or software. ISO has published in 2014 the ISO 29110 Systems Engineering Basic profile (ISO 2014) and the Systems Engineering Entry profile in 2015 (ISO 2015b). The set of deployment packages, developed by the INCOSE VSE WG, to help implement the Basic profile, served as the basis to develop a publicly licensed Training Kit. The first of those Training Kits, covering the Requirements Engineering DP, teaches how an Eclipse Requirements Management Framework based tool can be used to implement the RE DP within a VSE. Being publicly licensed, the Training Kit can be adapted very easily to fulfill specific VSE needs.

Once the software engineering ISO 29110 Intermediate and Advanced profiles are ready, work will start on the two corresponding systems engineering profiles for VSEs. In parallel, a set of systems engineering Deployment Packages and corresponding Training Kits will be developed and deployed to support those profiles. PolarSys collaborates with the INCOSE VSE WG to provide such Training Kits that leverage the example of the PolarSys Rover demo. In 2016, PolarSys plans to deliver Training Kits to demonstrate complete Systems Engineering of the Rover with Papyrus SysML and with Capella. These training kits will demonstrate document generation as well as code generation.

Additional Information

The following Web site provides more information, as well as articles by WG24 members and deployment packages for software and systems engineering:

<http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html>

References

(Figueiro et al 2008) Figueiredo, P. Silveira, G., Sbragia, R., Risk sharing partnerships with suppliers: the case of Embraer, University of Sao Paulo, Faculty of Economics, Accounting and Business Administration, Sao Paulo, Brazil, published in the Journal of Technology Management and Innovation, Volume 3, Issue 1. 2008.

Available at: <http://www.jotmi.org/index.php/GT/article/view/art70>

¹⁶ <https://github.com/jastram/teaching/>

(Garcia 2015) Garcia, L., Laporte, C.Y., Arteaga, J., Bruggmann, M., Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru, Software Quality Professional Journal, ASQ, vol. 17, no. 2, 2015, pp. 16-29

(ISO 2011) ISO/IEC/IEEE 29148:2011, Systems and software engineering - Life cycle processes - Requirements engineering. Geneva, Switzerland: International Organization for Standardization (ISO), 2011.

(ISO 2014) ISO/IEC TR 29110-6-5-2:2014 - Systems and Software Engineering – Systems Engineering Lifecycle Profiles for Very Small Entities (VSEs) - Management and engineering guide: Generic profile group: Basic profile, Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission. Available at no cost from ISO at: http://standards.iso.org/ittf/PubliclyAvailableStandards/c063371_ISO_IEC_29110-5-6_2_2014.zip

(ISO 2015a) ISO/IEC/IEEE 15288:2015, Systems and software engineering - System life cycle processes, Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission, 2015.

(ISO 2015b) ISO/IEC TR 29110-6-5-1:2015 - Systems and Software Engineering – Systems Engineering Lifecycle Profiles for Very Small Entities (VSEs) - Management and engineering guide: Generic profile group: Entry profile, Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission. Available at no cost from ISO at: <http://standards.iso.org/ittf/PubliclyAvailableStandards>

(ISO 2015c) ISO/IEC/IEEE 15289:2015, Systems and software engineering - Content of life cycle process information products (Documentation), Geneva, Switzerland: International Organization for Standardization/International Electrotechnical Commission.

(ISO 2016) ISO/IEC TR 29110-1:2016, Systems and software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 1: Overview. Geneva, Switzerland: International Organization for Standardization (ISO), 2016. Available at no cost from ISO at: <http://standards.iso.org/ittf/PubliclyAvailableStandards>

(Land 1997) Land, S.K., Results of the IEEE Survey of Software Engineering Standards Users, Software Engineering Standards Symposium and Forum, 1997. Emerging International Standards. ISESS 97, Third IEEE International, 1-6 June 1997 Page(s):242 – 270

(Laporte 2011) The Development and Experimentation of an International Standard for Very Small Entities Involved in Software Development, International Council on Systems Engineering (INCOSE) Workshop, January 29th 2011, Phoenix, Arizona.

(Laporte 2012) Laporte, Claude, Y, Fanmuy, Gauthier, Ptack, Ken, The Development of Systems Engineering International Standards and Support Tools for Very Small Enterprises, 22nd Annual International Symposium of the International Council on Systems Engineering, Rome, July 9-12, 2012. Available at:

http://www.etsmtl.ca/Professeurs/claporte/documents/publications/LAPORTE_INCOSE_2012.pdf

(Laporte 2013a) Laporte, C.Y., Séguin, N., Villas Boas, G., Seizing the benefits of software and systems engineering standards, ISO Focus+, International Organization for Standardization, February 2013, pp 32-36.

http://www.iso.org/iso/home/news_index/iso_magazines/isofocusplus_index.htm

(Laporte 2013b) Laporte, C.Y., Chevalier, F., Maurice, J.-C., Improving Project Management for Small Projects, ISO Focus+, International Organization for Standardization, February 2013, pp 52-55.

http://www.iso.org/iso/home/news_index/iso_magazines/isofocusplus_index.htm

(Laporte 2013c) Laporte, C.Y., O'Connor, R., Fanmuy, G., International Systems and Software Engineering Standards for Very Small Entities, *CrossTalk - The Journal of Defense Software Engineering*, May/June 2013, Vol. 26, No 3, pp 28-33. Available at:

http://www.etsmtl.ca/Professeurs/claporte/documents/publications/Crosstalk_May_2013_Laporte.pdf

(Laporte 2014a) Laporte, Claude Y., Houde, R., Marvin, J., Systems Engineering International Standards and Support Tools for Very Small Enterprises, Paper to be presented at the 24th Annual International Symposium of INCOSE (International Council on Systems Engineering), Las Vegas, U.S., June 30-July 3, 2014. Available at:

<http://www.etsmtl.ca/Professeurs/claporte/documents/publications/INCOSE-2014.pdf>

(Laporte 2014b) Laporte, C.Y., O'Connor, R., A Systems Process Lifecycle Standard for Very Small Entities: Development and Pilot Trials, 21st European Software Process Improvement Conference (Euro SPI 2014), Luxembourg, June 25-27, 2014.

(Laporte 2015a) Laporte, C.Y., Chevalier, F., An Innovative Approach to the Development of Project Management Processes for Small-scale Projects in a large Engineering Company, 25th Annual International Symposium of INCOSE (International Council on Systems Engineering), Seattle, US, July 13-16, 2015. Available at:

http://www.etsmtl.ca/Professeurs/claporte/Publications/INCOSE-2015_Tetra_Tech.pdf

(Laporte 2015b) Laporte, C.Y., Houde, R., Open Source Systems Engineering Guides, Deployment Packages and Support Tools for Very Small Enterprises – A Case Study, 25th Annual International Symposium of INCOSE (International Council on Systems Engineering), Seattle, US, July 13-16, 2015.

Available at: http://editionweb.etsmtl.ca/Professeurs/claporte/Publications/INCOSE-2015_29110_Open-Source.pdf

(Moll 2013) Moll, R., Being prepared – A bird's eye view of SMEs and risk management, ISO Focus +, Geneva, Switzerland: International Organization for Standardization, February 2013. Available at no cost at:

http://www.iso.org/iso/home/news_index/iso_magazines/isofocusplus_index/isofocusplus_2013/isofocusplus_2013-02.htm

Biography

Dr. Claude Y. Laporte has been a professor since 2000 at the École de technologie supérieure (ÉTS), an 9000-student engineering school, where he teaches software engineering. His research interests include software process improvement in small and very small enterprises, as well as software quality assurance. He has worked in defense and transportation enterprises for over 20 years. He received a Master's degree in Physics from the Université de Montréal, a Master's degree in Applied Sciences from the École Polytechnique de Montréal and a Ph.D. from the Université de Bretagne Occidentale (France). In addition, he was awarded an honorary doctorate by the *Universidad de San Martin de Porres* (Peru) in 2013. He is the Editor of ISO/IEC JTC1 SC7 Working Group 24, tasked to develop ISO/IEC 29110 life cycle standards and guides for Very Small Entities. He is the Co-chair of the INCOSE Systems Engineering for Very Small Entities WG. He is a member of INCOSE, IEEE, PMI and a member of the professional association of engineers of the Province of Québec (*Ordre des ingénieurs du Québec*). He is the co-author of two French books on software quality assurance published in 2011 by Hermes Science-Lavoisier and one English textbook, on the same topic, published by John Wiley and Sons in 2016.

Web site address: <http://profs.etsmtl.ca/claporte/English/index.html>

Ronald Houde is a Senior Systems Engineering Specialist with Mannarino Systems & Software. He has over 30 years of experience in government and commercial safety- and mission-critical software and systems engineering. Employers and customers have included world leaders such as Esterline/CMC Electronics, Lockheed Martin, Bombardier Aerospace, CAE, BPR Énergie, Hydro-Québec, the Canadian Department of National Defence and the U.S. Army. His main areas of employment have been in the design, development, integration and installation of safety-critical software systems and programmable electronic devices. His clients in both industry and government benefit from his skills in training and mentoring, complex problem analysis and solving, and communication. He has also been involved in technical training and university and continuing education programs since the early 1990s. He is experienced in the conduct, management and continuous process improvement of software and systems engineering activities covering the entire life-cycle of operational, mission support and safety-critical software systems. He is an experienced instructor to technical military and civilian audiences, having developed and taught numerous Avionics, Avionics Databus and Requirements Engineering workshops.

LinkedIn: <https://ca.linkedin.com/in/ronald-houde-9895977>

Gaël Blondelle is Director of European Ecosystem Development at the Eclipse Foundation and manager of the PolarSys Working Group since 2013. Gaël Blondelle has accumulated specialized experience with Open Source communities since 2002. From 2010 to 2013, he has been manager of OPEES, a European project whose key result is the creation of PolarSys, the Eclipse Working Group dedicated to Open Source tools for Embedded Systems. In 1995, Gaël received an Engineering Degree in Software Engineering and Artificial Intelligence from ESIAL (now Telecom Nancy). He started working in the software industry in 1996, and since then, he has been working primarily in Telecommunications, IT and Embedded Systems industries with expertise in Java, Web Services, SOA and Embedded Systems in

large organization, consulting firms as well as a co-founder and CTO of a startup, PetalsLink that was producing the Open Source ESB Petals.

LinkedIn: <https://fr.linkedin.com/in/gblondelle>