

ISO/IEC 29110 Deployment Packages and Case Study for Systems Engineering: The "Not-So-Secret" Ingredients That Power the Standard

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

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

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

Copyright © 2016 by R. Houde, C. Y. Laporte and G. Blondelle, Permission granted to INCOSE to publish and use.

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. 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. The ISO/IEC 29110 systems engineering management and engineering guides were developed mainly from ISO/IEC/IEEE 15288 to address this new reality, to exploit the lean and efficient nature of VSEs and to adapt to their typical budget and resource constraints. By design, the management and engineering guide is supported by Deployment Packages (DP), the development of which was taken on by the INCOSE VSE Working Group. A DP is a set of artefacts designed to facilitate the implementation of the management and engineering guides of ISO/IEC 29110 by VSEs. In tune with the need for low cost and flexibility, Open Source software tools are emerging to support VSEs and provide a bridge with “Big League” development life-cycle toolsets. Finally, to make the deployment of ISO/IEC 29110 possible in VSEs, training packages, supported by relevant pilot projects help VSE personnel learn how to apply all of the above. This paper describes the Systems Engineering DP for Requirements Engineering (RE DP) and shows how it can be applied using the Autonomous Rover Case Study developed under the Eclipse Foundation Polarsys project.

Introduction

Industry recognizes the value of Very Small Entities (VSEs), i.e., enterprises, organizations (e.g. government agencies, not-for-profit 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 the 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, in order 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 began to participate more directly in investments and project development, thereby acquiring rights to their future sales income.” (Figueiredo 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 regulated sectors such as aerospace, medical devices, rail transportation, atomic energy and industrial processes, the transfer of responsibilities entailed the production of a range of process artefacts supporting the certification process 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 ISO 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 and mandated 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 (ISO 29110 hereon) Deployment Packages developed for VSEs. More specifically, the paper:

- Provides an overview of the ISO 29110 standards and guides for VSEs;
- Describes the set of nine (9) DPs covering the span of the systems engineering life-cycle;
- Describes in more detail the Requirements Engineering (RE) DP;
- Describes a Case Study that was selected and developed to illustrate how the ISO 29110 guides and DPs can be applied and
- Describes how the templates and checklists of the RE DP can be applied using low-cost Open Source tools originating from Eclipse Foundation Polarsys Project's Autonomous Rover Case Study to capture stakeholder and system requirements using the Eclipse Requirements Management Framework (RMF).

Overview of ISO 29110 Systems Engineering Standards and Guides

An ISO 29110 "Profile Group" consists of a four-stage roadmap, a stage commonly being referred to as a "profile" and the profiles earmarked as Entry, Basic, Intermediate and Advanced. The first Profile Group developed by ISO JTC1/SC7 Working Group 24, known as the Generic Profile Group, applies to VSEs that develop general purpose systems that are neither security- nor safety-critical.

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

Systems Engineering Deployment Packages

The INCOSE VSE Working Group defined a set of guidelines explaining in more detail the management and engineering processes defined in the Basic profile. The Basic profile targets VSEs developing a single application by a single work team. These guidelines are freely accessible to VSEs on the Internet as a collection of DPs. A DP is a set of artefacts developed to facilitate the implementation of ISO 29110 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 DP is illustrated in Table 1.

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

Table 1: 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 and steps
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 1 illustrates the set of SE DPs for the Basic profile, which is available on the Internet² and on the INCOSE VSE page.

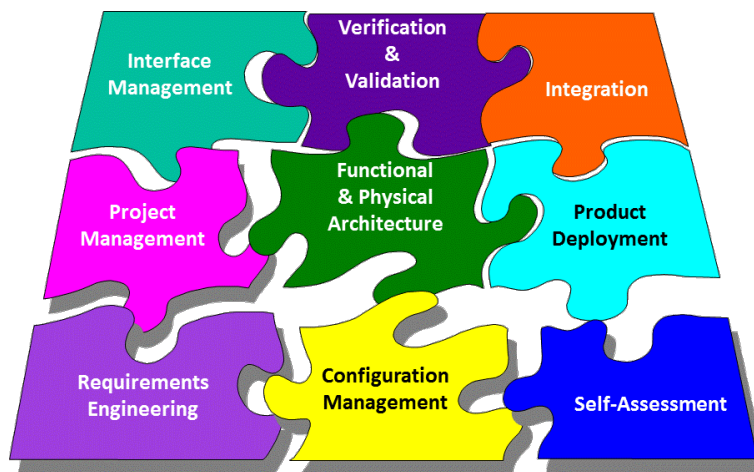


Figure 1. Deployment Packages for the SE Basic Profile

² <http://profs.etsmtl.ca/claporte/English/VSE/index.html> (Deployment Packages)

Requirements Engineering Deployment Package (RE DP)

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

- the capture of Acquirer/Stakeholder Requirements;
- the capture 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 a VSE and the specification assembling and structuring the collection of individual requirements. The checklists apply international standards such as ISO/IEC 29148, about requirements engineering (ISO 2008), and the INCOSE Guide for Writing Requirements.

Open Source Tools Available 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 financial or technical support burden of “Big League” tools and have been relegated to using word processing and spreadsheet tools to support system or software development, starting with managing requirements. Current commercially available requirements management tools add two further constraints that will often discourage VSEs: 1) the tool’s flexibility requires that the RM tool 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

(RMF)^{3,4}, the Object Management Group (OMG) Requirements Interchange Format (ReqIF)⁵ and the support of such initiatives as the Eclipse Foundation Polarsys⁶ project, 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. However, those tools are often generic in nature and require some tailoring which is addressed in the RE Deployment Package for the benefit of VSEs.

RMF is a framework for working with textual requirements, structured as ReqIF models. RMF uses ReqIF natively, allowing an organization to exchange requirements with most of the sophisticated 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 partners and upper tier clients/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 database. 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.

Alternately, if the VSE opts to apply a Model-Based Systems Engineering (MBSE) approach, the requirements can be captured in SysML Requirements Diagrams using an environment such the Eclipse Foundation Papyrus SysML⁷ modeling tool. Papyrus is an open source, model-based development tool that provides an industrial-strength, community-backed solution for system and software developers who demand a standards-compliant, reliable, modular, extensible, and configurable modeling tool. Papyrus is also becoming increasingly popular as the core capability of commercially available environments and distributions. The use of Papyrus, however, requires that ReqIF requirements be imported into the Eclipse Modeling Framework (EMF)⁸. This capability will be demonstrated in the Functional & Physical Architecture DP portion of the Autonomous Rover Case Study.

The Autonomous Rover Case Study

In order to demonstrate how DPs can be applied on a representative system application, the INCOSE VSE Working Group set out to identify a suitable Case Study. In the course of the search, the team discovered the Autonomous Rover Case Study defined by the Eclipse Foundation Polarsys Project⁹. The VSE Working Group then decided to join forces with the Polarsys project and develop a version of the Autonomous Rover Case Study that would serve to fulfill the need¹⁰.

³ <http://www.eclipse.org/rmf/>

⁴ Under the leadership of Michael Jastram

⁵ <http://www.omg.org/spec/ReqIF/>

⁶ <http://www.polarsys.org>

⁷ <https://www.polarsys.org/solutions/papyrus>

⁸ <https://eclipse.org/modeling/emf/>

⁹ https://polarsys.org/wiki/PolarSys_Rover_Demo

¹⁰ https://polarsys.org/wiki/Systems_Engineering_the_Autonomous_Rover

The Autonomous Rover is a sample application meant to develop a small robot, identified as the Autonomous Rover, which would assist Emergency Hazardous Material (HazMat) Response Technicians map and assess the situation in a theatre where the presence of a risk to human life is suspected.

As a seed, helping to frame the Autonomous Rover solution space, the Operational Concept (OpsCon) document for the project specifies that the stakeholders selected “the Dagu Electronics Rover 5 chassis with 2 motors and encoders as a standard chassis for all their small robot and rover applications.”¹¹

All files necessary to perform the case study are deposited in a GitHub project¹². The repository is openly accessible and usable under an Apache 2.0 license. It contains:

- the Bill-Of-Materials (BOM) for the material and electronic components of the Autonomous Rover, in both Arduino and Raspberry Pi implementations;
- a RMF ReqIF Template containing blank Stakeholder, System Requirements and requirements traceability links databases;
- RE DP template documents for the Stakeholder Requirements and System Requirements documents, in Open Document Template (.opt) format;
- the RE DP Checklists in the form of Open Document Templates (.opt)
- the reference Stakeholder Requirements and System Requirements documents for the Autonomous Rover in Open Document format (.opd);

Figure 2 presents an excerpt of the Autonomous Rover Stakeholder Requirements Specification.

¹¹ Stakeholder Requirements Specification (StRS) for the Autonomous Rover. The document is available from the Autonomous Rover Case Study GitHub Repository.

¹² <https://github.com/perdlespedales/ISO29110-Polarsys-Rover>

StRS for the Autonomous Rover

6.2 Operational scenario

6.2.1 Deployment to the Theatre of Operations

The Hazardous Material Response Team will dispatch to the theatre of operations by road or air. The Autonomous Rover, batteries (including spare batteries and battery charger, if required) sensor payloads and critical spare parts will be transported in the air transport case.

Because it is never known where a mission can be dispatched from, it is necessary to carry all sensors along with the Autonomous Rover.

6.2.2 Autonomous Rover Preparation

Once having reached the site where the mission will be performed, the Emergency Response Technician will assemble the Autonomous Rover, adjust the chassis height, mount the sensors needed for the mission, power-up the Autonomous Rover and perform a power-up check and battery level verification.

The Autonomous Rover should either be able to automatically recognize connected sensors or provide a means for the HazMat Response Technician to enter the sensor configuration once they are mounted.

6.2.3 Theatre Mapping and Data Collection

Once the HazMat Response Technician is satisfied the Autonomous Rover is ready for the mission, he/she will position the Autonomous Rover for the mission in one of two ways:

- In Phase 1 deployment, the technician will position by hand the Autonomous Rover at the starting point of the zone to be explored;
- in subsequent deployment phases, the technician will have the option to use a tablet remote control device to position the rover at the starting point of the zone to be explored.

During the mission, since the technician will likely not have a direct sight to the Autonomous Rover, it is not desirable that it be possible to steer the robot remotely in any way.

During the mission the Autonomous Rover should adopt an avoidance and evasion strategy if it encounters a threat or danger. For example, if the rover is capable of detecting a fire and encounters one it will steer to stay at a distance of the fire. Similarly, if is capable of detecting a water hazard, it will avoid the water hazard rather than attempt to go through it.

Figure 2 – Sample Stakeholder Requirements

For each of the Steps defined in the RE DP, and shown in Figure 3, the GitHub repository provides:

- the RE artefacts as they should be at the beginning of the Step; and
- the RE artefacts as they could be once the Step has been completed.

Contents [hide]	
1	Introduction
2	Activities - Phase 1
2.1	SR.1.1 – Review Project Plan with the Work Team members
2.2	SR.2.1 - Elicit Acquirer and other stakeholders requirements and analyze system context
2.3	SR.2.2 - Review Stakeholders Requirements Specifications with PM
2.4	SR.2.3 - Baseline Stakeholders Requirements Specification with the Acquirer and Stakeholders
2.5	SR.2.4 - Capture System Requirements and Interfaces
2.6	SR.2.6a - Verify and obtain Work Team (WT) agreement on the System Requirements Specification
2.7	SR.2.7 - Validate that System Requirements Specification satisfies Stakeholders Requirements Specification
2.8	SR.2.8a - Define or update traceability between Requirements (System to Stakeholder)
3	Activities - Phase 2
3.1	SR.2.5 - Capture System Elements and Interface Requirements
3.2	SR.2.6b - Verify and obtain Work Team (WT) agreement on the System Elements Requirements Specifications
3.3	SR.2.8b - Define or update traceability between Requirements (System Element to System)
4	Navigation Links

Figure 3 Autonomous Rover Case Study Requirements Engineering Steps

The sample artefacts offered at each step of the Case Study do not represent the only possible solution to the stated stakeholder needs, but are: 1) representative; and 2) provides for the continuity of the evolution of the Case Study.

The following Figures show how Model-Based Use Case Analysis can be applied to carry-out Requirements Analysis. The analysis is carried out using the Papyrus SysML modelling tool. In Figure 4, a Use Case Diagram depicts how the Autonomous Rover interacts with external Actors identified in the OpsCon document. This activity corresponds to Task SR.2.1 above.

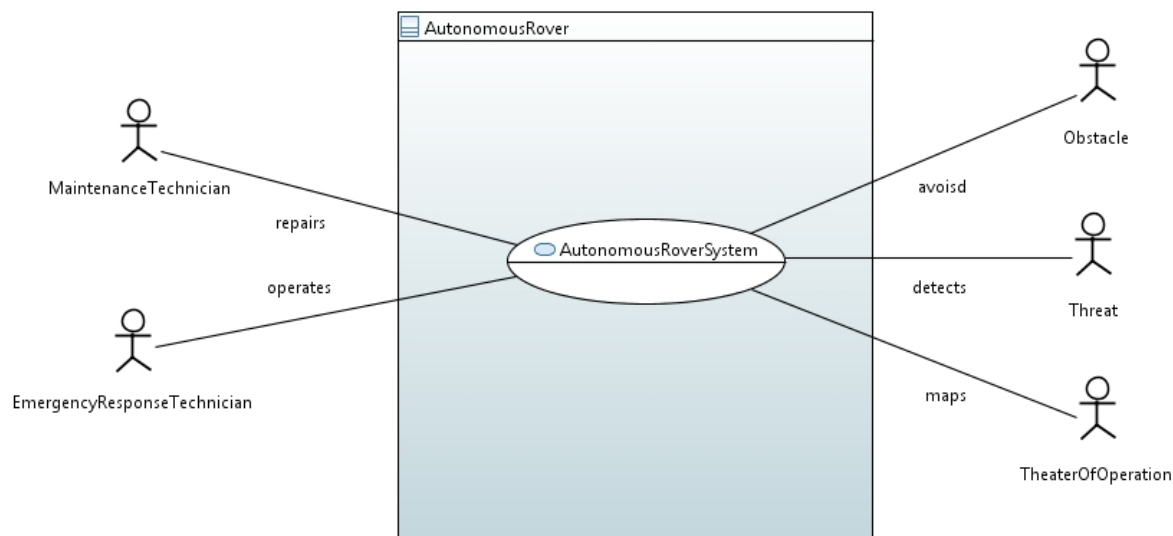


Figure 4 Autonomous Rover System Context Diagram

The Systems Engineer then, using the Autonomous Rover Stakeholder Requirements, identifies Use Cases for the Rover. Figure 5 shows the Rover's Main Use Case Diagram.

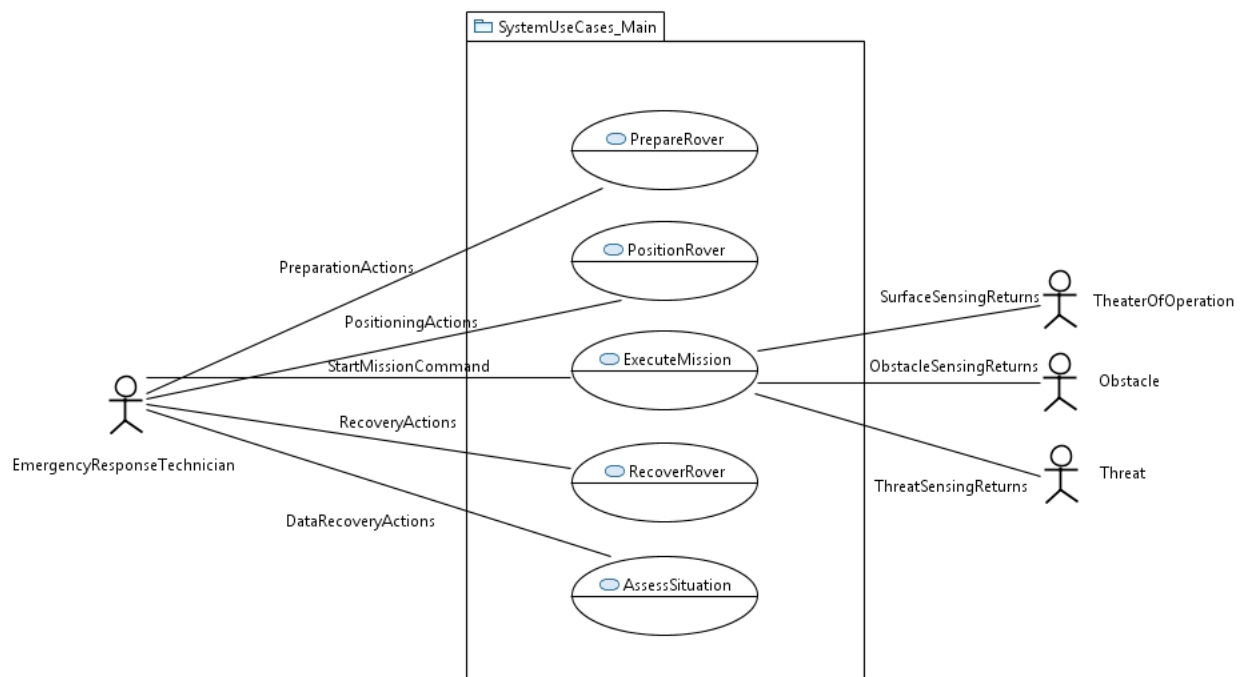


Figure 5 Autonomous Rover Main Use Case Diagram

Finally, using a combination of Activity Diagrams, Sequence Diagrams and State Machine Diagrams, the Systems Engineer ensure operational scenarios are understood and system requirements from the System Requirements Document provide a complete, coherent and unambiguous set of System Requirements traceable to the Stakeholder Requirements. Figure 6 shows how a State Machine Diagram is used to develop an understanding of the Execute Mission Use Case.

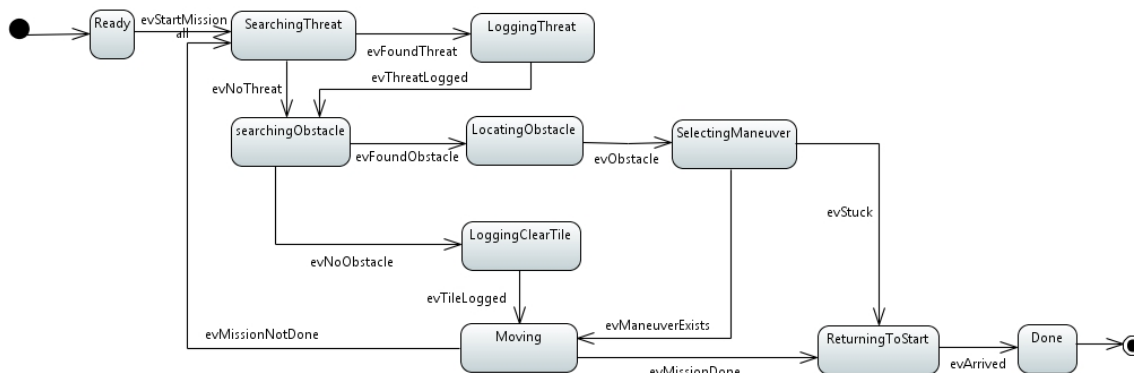


Figure 6 Execute Mission State Machine Diagram

In the Case Study, students are made to realize using these analysis techniques that System Requirements are missing for the “Assess Situation” Use Case.

The Autonomous Rover Prototype

Prototypes are a powerful tool to help understand operational concepts, possible system solutions and architectures and the feasibility of candidate solutions. Figures 7 and 8 show a completed Autonomous Rover prototype that uses an Arduino Mega 2560 as its programmable micro-controller. In the course of preparing the Case Study material, it was determined that an Arduino Uno micro-controller configuration would likely not have sufficient memory, timers and I/O channels to implement a solution for the Autonomous Rover.

The Autonomous Rover prototype developed for the Case Study was entirely built from COTS components identified in the Autonomous Rover BOM, many of which are Open Source, including:

- the Dagu Electronic Rover 5 chassis, as specified in the Polarsys Case Study scope;;
- a Dagu Electronics Explorer Printed Circuit Board interface including:
 - 2x FET “H” bridge motor drivers rated at 4A with current sensing
 - Servo power supply – 5.4V rated at 3A assuming the use of 7.2V battery
 - 5V LDO regulator rated up to 1.5A with suitable heatsinking
 - 3.3V LDO regulator rated at 500mA
 - Built in trickle charge circuit for NiMh or NiCd batteries.
 - 4x IR analog/digital corner-mounted obstacle sensors
 - 2x individual power rails (V1, V2) that can be connected to any voltage
 - Mounting holes for hardware including Arduino and Arduino Mega PCBs.
- 1x AdaFruit CC3000 Wi-Fi Shield (for Phase 2 and 3 capabilities)
- 2x small bread boards with mounting hardware.
- 1x 2 degrees-of-freedom pan/tilt turret with two miniature servos and mounting hardware
- The following sensors mounted on the turret
 - 1x IR compound eye
 - 1x Ultrasound rangefinder
- A Rover body-mounted explosive fumes detector

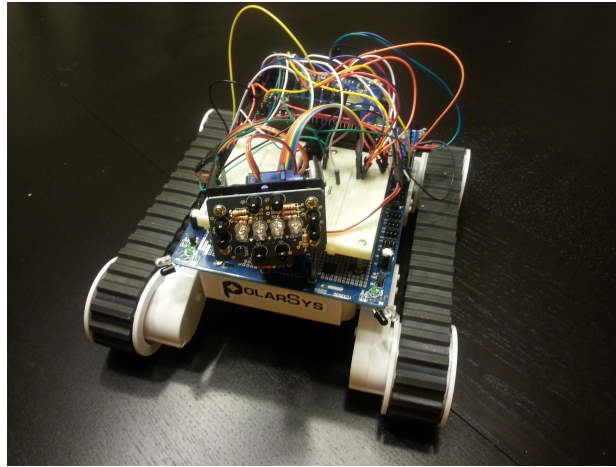


Figure 7 - Autonomous Rover Prototype with IR Compound Eye (Front View)

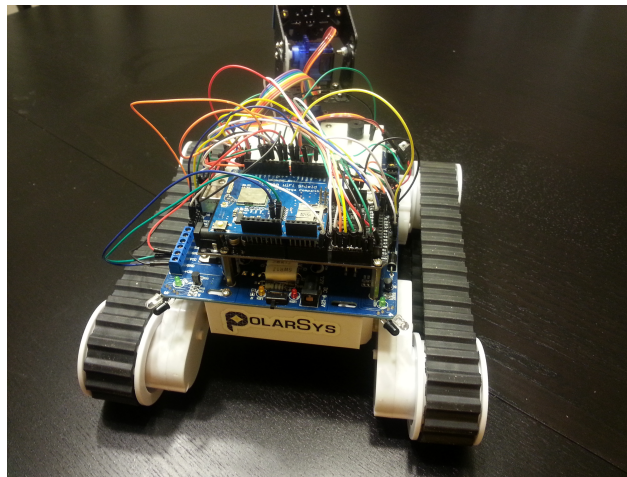


Figure 8 - Autonomous Rover Prototype with Arduino Mega 2560 (Rear View)

Alternate Autonomous Rover prototypes were also constructed using Raspberry Pi B+ and BeagleBone Black computer cards. The three controller configurations, in combination with the dual turret sensors and explosive fumes sensor provide a means of supporting trade studies in the Functional and Physical Architecture DP segment of the Case Study.

Figure 9 shows a complete Phase 2 prototype Autonomous Rover system, battery charger, Android remote control tablet, field repair kit and air transport case.



Figure 9 – The Autonomous Rover System Prototype

In parallel with the development of the Systems Engineering Autonomous Rover Case Study, an Autonomous Rover Software Engineering Case Study, using the management and engineering guide of the ISO 29110 Software Engineering Basic Profile (ISO 2011b), will demonstrate how System-to-Software Hand-off would typically be realized and how an end-to-end system-to-software development life-cycle can be realized. Ultimately, the Autonomous Rover Case Study can be extended to cover such disciplines as mechanical engineering (e.g. the development of a 3D printable electronics and sensor housing to mount on the Rover 5 chassis), electronics engineering (e.g. the conversion of the prototype electronics design to a custom made-to-fit electronics and Printed Circuit Board design fitting in the above electronics and sensor housing).

This Case Study will be offered to any interested organization such as INCOSE Chapters as a training kit to help their members and local VSEs implement ISO 29110 and increase their competence in systems engineering. The Case Study will also be presented to academia.

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.

The INCOSE VSE Working Group is in the process of completing the development of all 9 DPs. In parallel, in cooperation with the Eclipse Foundation's Polarsys Project an Autonomous Rover Case Study is being expanded so that it ultimately covers all the DPs of the Basic Profile.

The software Intermediate profile is used as the foundation for the development of the SE Intermediate profile. Once the software engineering ISO 29110 Advanced profile is ready in 2017, work will start on the corresponding Advanced systems engineering profile for VSEs. In parallel, a set of systems engineering Deployment Packages and corresponding Training Kits will be developed and deployed to support those SE profiles.

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>

(ISO 2008) ISO/IEC/IEEE 29148:2011, Systems and software engineering -- Life cycle processes -- Requirements engineering, Geneva, Switzerland: International Organization for Standardization (ISO), 2008.

(ISO 2011a) ISO/IEC TR 29110-1:2011, Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 1: Overview. Geneva, Switzerland: International Organization for Standardization (ISO), 2011. Available at no cost from ISO at:

http://standards.iso.org/ittf/PubliclyAvailableStandards/c051150_ISO_IEC_TR_29110-1_2011.zip

(ISO 2011b) ISO/IEC TR 29110-5-1-2:2011, Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-1-2: Management and engineering guide - Generic profile group: Basic profile, International Organization for Standardization/International Electrotechnical Commission: Geneva, Switzerland. Available at no cost from ISO at:

http://standards.iso.org/ittf/PubliclyAvailableStandards/c051153_ISO_IEC_TR_29110-5-1_2011.zip

(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.

(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. Available at: 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. Available at: 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_Laport e.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 2015) Laporte C.Y., Houde, Ronald, Open Source Systems Engineering Guides, Deployment Packages and Support Tools for Very Small Enterprises – A Case Study, INCOSE International Symposium 2015, Seattle, WA, USA, July 2015

(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.

Biographies

Ronald Houde is a Senior Systems Analyst with Mannarino Systems & Software and a delegate of Canada on the ISO JTC1/SC7 Working Group 24. 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 profile: <https://ca.linkedin.com/pub/ronald-houde/7/597/989>

Dr. Claude Y. Laporte has been a professor since 2000 at the École de technologie supérieure (ÉTS), a 9,000-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 Martín de Porres* (Peru) in 2013. He is the Project 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, to be published by John Wiley and Sons in 2016.



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

LinkedIn profile: <https://www.linkedin.com/pub/claude-y-laporte-eng-ph-d/2/37b/bb0/fr>

Gaël Blondelle is the Director of European Ecosystem Development at the Eclipse Foundation. Gaël 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 2013, Gaël joined the Eclipse Foundation where he is in charge of the PolarSys Working Group and the development of the European ecosystem. 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 start-up he co-founded.



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