

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

Towards Configurable ISO 29110-compliant Software Development Processes for Very Small Entities

Boucher, Quentin; Perrouin, Gilles; Deprez, Jean-Christophe; Heymans, Patrick

Published in:

Proceedings of the 19th European System, Software & Service Process Improvement & Innovation Conference (EuroSPI'12), Vienna, Austria

Publication date:

2012

Document Version

Early version, also known as pre-print

[Link to publication](#)

Citation for published version (HARVARD):

Boucher, Q, Perrouin, G, Deprez, J-C & Heymans, P 2012, Towards Configurable ISO 29110-compliant Software Development Processes for Very Small Entities. in *Proceedings of the 19th European System, Software & Service Process Improvement & Innovation Conference (EuroSPI'12), Vienna, Austria*. Springer, pp. 169-180, 19th European System, Software & Service Process Improvement & Innovation Conference (EuroSPI'12), Vienna, Austria, 25/06/12.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Towards Configurable ISO/IEC 29110-compliant Software Development Processes for Very Small Entities

Quentin Boucher¹, Gilles Perrouin¹, Jean-Christophe Deprez², and Patrick Heymans^{1,3}

¹ PRECISE Research Centre, University of Namur, Belgium

² Centre of Excellence in Information and Communication Technologies, Belgium

³ INRIA Lille-Nord Europe, Université Lille 1 – LIFL – CNRS, France

{qbo, gpe, phe}@info.fundp.ac.be, jean-christophe.deprez@cetic.be

Abstract. Using ISO/IEC 29110, very small entities (VSEs) can perform a step-wise increment of their software process by switching between the different ISO/IEC profiles. However, ISO/IEC 29110 provides no guidance on how to switch between profiles incrementally, other than resorting to costly software development process experts unaffordable for VSEs. To address this shortcoming, this paper shows how to model the variability of currently available ISO/IEC 29110 profiles in an integrated and configurable workflow with illustration on the Requirements Engineering (RE) activity. This workflow is linked to a questionnaire used to support automated process configuration. Thereby, the user can easily derive the ISO/IEC-compliant processes to switch between profiles incrementally. The feasibility of this approach is shown using open-source workflow management tools Synergia and YAWL.

Keywords: Workflow Management, Configuration, ISO/IEC 29110, Requirements Processes

1 Introduction

From the first versions of Software CMM [1] and ISO/IEC 12207 [2], reference process models for software development have attracted a lot of interest over the past decade. However, they have failed to become accepted by Very Small Entities (VSEs), i.e., enterprises, organizational units or projects composed of 25 people or less [3]. The major criticism of the aforementioned standards relates to their excessive complexity, and thus their inapplicability to contexts where resources are extremely limited. The second recurring criticism is the technical jargon used in standards [4]. Nevertheless, VSEs remain interested by ISO certification [4]. To fill this gap, ISO/IEC recently published the ISO/IEC 29110 standard [3], which provides adapted development processes for VSEs.

Despite notable effort to make ISO/IEC 29110 more applicable to VSEs than its predecessors, considerable time and resources to understanding and applying the standard is still needed often requiring intervention from software process consultants. Moreover, if certification is the target—which seems to be the case for many VSEs [4]—records of systematic application of the standard must be kept and exhibited upon demand to certification authorities. The adoption barrier thus remains high.

To alleviate the difficulties of applying the ISO/IEC 29110 standard, the concept of *deployment package* has been introduced [5]. Deployment packages are additional documents, task cards [6], detailed process descriptions and templates [7]. Although they improve the understanding of the standard, they are of little help for the concrete realization of reference models, and do not guarantee compliance throughout the project lifecycle. Furthermore, reference models still need to be tailored to the specific operational needs of the adopting VSE.

In this paper, we propose to address the above problems by applying *configurable workflow*¹ [8] concepts and technology. First, we model the *Basic* and *Entry* profiles of the ISO/IEC 29110 *Software Requirements Analysis* sub-process as a single, integrated, workflow. This workflow exhibits the commonalities and differences between the two profiles and allows a fine-grained tailoring of the activities and artefacts. Tailoring is realized by simply asking the user to answer a set of questions [9]. Answers to these questions are used to automatically derive a correct and ISO/IEC-compliant workflow that can be integrated within widespread workflow engines. The approach is currently being implemented using the Synergia tool suite [10].

The paper is organized as follows. Section 2 presents the ISO/IEC 29110 standard. It also relates the configurable workflow approach to principles of method engineering and introduces our research methodology. Section 3 describes our application of the latter concepts and the implementation. In Section 4, our contribution is discussed. Finally, Section 5 wraps up the paper and presents some on-going and future developments.

2 Background and Related Works

2.1 ISO/IEC 29110

In 2011, ISO/IEC published a first version of a five-part standard, named “ISO/IEC 29110 - Lifecycle Profiles for Very Small Entities” [3]. Parts 1 and 5 of ISO/IEC 29110 target a VSE audience. Part 1 presents an overview of the standard whereas Part 5 describes the major software development lifecycle processes for VSEs. The three other parts present mandatory information for all standards but do not necessarily need to be understood by VSEs. Part 2 introduces the general framework for developing current and future profiles; Part 3 discusses the assessment dimension; Part 4 establishes a correspondence between elements of ISO/IEC 29110 and their counterparts in ISO/IEC 12207.

From inception, the intent is to create multiple profiles to define a process improvement ladder made of four rungs. Based on its current process maturity, a VSE can then start at the appropriate rung (ranging from *Entry* to *Advanced*) by setting up the software process for that rung before climbing to the next rung. In the 2011 version of ISO/IEC 29110, only one profile was included in the standard: the *Basic* profile (rung 2). At the moment, an internal version of the *Entry* profile (rung 1) also exists and will be officially published in 2012.

The profiles defined in ISO/IEC 29110 use the following main concepts:

¹ We use the term “workflow” to denote an executable process model.

- A *software process* is composed of a set of objectives. An enterprise is said to have a software process in place if an auditor determines that the process’ objectives are reached by the work performed by the enterprise.
- A *process task* describes the work that staff members with particular roles need to perform using a specified set of inputs to create the expected set of outputs.
- A *process activity* groups a set of related process tasks. A process activity only appears in one process and a task only appears in a single process activity.

The *Entry* and *Basic* profiles of ISO/IEC 29110 are composed of two processes: project management (PM) and software implementation (SI). The *PM-Entry* and *SI-Entry* are actually lighter than their counterparts in the *Basic* profile. Pragmatically, the *Entry* profile was created by editing out portions of the *Basic* profile. In some cases, tasks were removed altogether. In others, the work required to perform a task was simplified and eventually simplified tasks were merged.

2.2 Method Engineering

Method Engineering (ME) is the discipline of designing, constructing and adapting methods, techniques and tools for the development of information systems [11]. *Situational* method engineering refers to the customization of methods for the particular project and context at hand. Situational method engineering generally proceeds by combining method fragments reused from a common “methodbase” [12]. Sometimes only one fragment of the method is changed, or *incremental* ME. Naturally, this fragment is usually changed to improve the performance of the overall method by finetuning it to a specific situation. This makes the link between incremental ME and software process improvement obvious [13–15]. Another way to deal with flexibility is to introduce variation points in the method itself. This is the approach followed by ISO/IEC with their lifecycle profiles. Then, it is possible to perform *method configuration* [16] to tailor an existing base method. Thus, ME and method configuration form a conceptual framework to define and tailor software engineering processes such as defined by ISO/IEC.

Over the years, several Computer Assisted Method Engineering (CAME) tools, such as MetaEdit+ [17], have been developed to assist method engineers tuning up method to the needs of a project. However, as noted by Cervera et al. [18], besides MetaEdit+, very few tools reached industrial maturity. Furthermore, industry usually thinks that the costs of applying situational ME is perceived as being larger than using an off-the-shelf method [12].

If we focus on VSEs, the target of our approach, further issues arise. One is the cost of such CAME tools. VSEs generally have limited financial resources and prefer to invest in software development tools than in tools guiding their development process. As a result, certification and self-assessment of ISO/IEC 29110 compliant VSEs is challenging. Providing an affordable approach to software lifecycle management is currently the focus of the NAPLES project². Additionally, VSEs hardly have ME skills internally: subtleties of ISO/IEC 29110 profiles’ combinations may be overlooked (unnecessarily forbidding relevant method configuration) or, on the contrary, important constraints may be ignored allowing incorrect application of the standard.

² <http://www.cetic.be/NAPLES,1162>

Therefore, our research relies on open-source solutions and strives to provide a reusable method configuration approach based on the ISO/IEC 29110 to reduce the ME effort to its minimum. Our approach relies on workflow configuration which is detailed in the next paragraphs.

2.3 Configurable Workflows

Workflow Management Systems (WFMS) help realizing processes in an automated way. Powerful open-source WFMS, such as YAWL [19] or Bonita³ make workflow-based applications development accessible to all. A first, naïve approach, to workflow-based ISO/IEC 29110 compliance would be to model each profile as an independent workflow and let the users pick the one they want to apply. There are three drawbacks to this. First, several activities and tasks of the ISO/IEC 29110 are common between profiles. Duplicating them across multiple workflows makes development and maintenance more costly. Second, as mentioned in Section 2.1, profiles are not mutually exclusive: an *Entry* level development process consists of activities and tasks also specified in the *Basic* profile. Finally, even if WFMS simplify the task of developing and customizing workflows, relying on a purely manual configuration would also be error-prone and would necessitate a thorough knowledge of the standard to guarantee the compliance of the resulting configured workflow. A more flexible and user-friendly approach should therefore improve the adoption by VSEs.

The concept of *configurable workflow* [8, 10] proposes a product-line [20] approach to workflow modelling. The principle is simple: a *domain* workflow model representing all legal workflow variants is defined expressing all variation points explicitly. Those variation points are then used during configuration where the *product* workflow model is produced by activating and deactivating activities. To facilitate configuration, questionnaire-driven configuration is being used [9]. User-oriented questions are associated to variation points, offering an additional abstraction layer on top of the variation points. The consistency of the answers given by the user is enforced by the configuration engine, in the same spirit as feature-based configuration in software product lines. Based on the answers, the initial all-variants domain workflow model is pruned from the deselected elements through an *individualisation* process. The individualisation process consists of transforming the domain workflow model into a valid product workflow model. This questionnaire-driven individualisation process is supported by the Synergia tool suite [10].

2.4 Research Methodology

Our goal was to explore the feasibility of configurable workflows to assist users applying the ISO/IEC 29110 standard. For this purpose, we first read the documentation of available profiles, *Entry* and *Basic*, and represented common as well as variable tasks into a single configurable workflow, so introducing variability points. This workflow was then validated by a member of subcommittee 7 of ISO/IEC-JTC1 defining the ISO/IEC 29110 standard. Then, we defined a question for each variation point of

³ <http://www.bonitasoft.com/>

one. Extracting the variability from the different profiles could thus be automated using tasks' IDs but should be validated by a human being to ensure that task's descriptions are the same. A complete description of the tasks appears in Table 1 where italic text denotes input and output products. The inclusion of a task in a profile is also displayed in the last two columns on the right side of the table. The corresponding YAWL workflow is depicted in Figure 1. There, tasks that are common to both profiles have a *red* (dark grey) background while *yellow* (light grey) tasks are specific to the *Entry* profile, and *orange* (grey) ones to the *Basic* profile. Triangles associated to tasks represent either a xor-join (mutually exclusive incoming transitions) or a xor-split (mutually exclusive outgoing transitions).

ISO/IEC 29110 allows combining tasks from both profiles. This means that a VSE can decide to follow the *Entry* workflow while borrowing some tasks from the *Basic* workflow, e.g. when a customer needs to perform some subprocesses in a different or more detailed manner. As long as all tasks of a given profile are included, compliance with this profile is guaranteed. Moreover, some tasks can be defined as optional in the standard for all or some profiles. For example, tasks SI.2.5, SI.2.6 and SI.2.7.b are optional because *Software user documentation* is optional in the ISO/IEC 29110 standard. To assist users with process configuration, we identified five questions which allow to select tasks from Figure 1:

- Q1 - *Are the requirements gathered into a repository without automated versioning (e.g. in a spreadsheet) ?* – “Yes” or “No”
- Q2 - *When do the correctness and testability of the requirements specification and its consistency with the product description have to be verified?* – “Once, when all requirements have been identified” or “Iteratively, until fully approved”
- Q3 - *Do you wish to keep track of the requirements verification and validation process?* – “Yes” or “No”
- Q4 - *Do you plan to produce a Software User Documentation?* – “Yes” or “No”
- Q5 - *Do you wish to incorporate the requirements specification as a baseline in the software configuration ?* – “Yes” or “No”

Those questions have been defined with the *Questionnaire Designer* tool of Synergia [10] (see Figure 2). There, green (dark grey) boxes correspond to our five questions and yellow (light grey) boxes to the answers, called “facts”. Each fact is associated to a question through a “MapQF” relationship. Dependencies between questions/facts, i.e. a question/fact is available if and only if another question/fact has been answered/selected, can also be defined but were not used in our example. The tool can then save the questionnaire in a XML format.

Then, Synergia's *C-mapper* tool is used to link questionnaire files and processes to allow configuration. Questions are then processed by *Quaestio* which interacts with the user and saves the answers. A negative answer to question *Q1* implies the selection of the task SI.2.2.b. The next question, *Q2*, has two possible answers which are mutually exclusive. The first answer is linked to task SI.2.3 as all requirements are collected before verifying them, while the second one is associated to task SI.2.2.d. Question *Q3* is directly mapped to task SI.2.4.a. Question *Q4* is associated to all tasks that include the software user documentation, namely SI.2.5, SI.2.6 and SI.2.7.b. The answer to the last question, *Q5*, will determine the selection (posi-

Table 1. ISO/IEC 29110 – List of tasks of the software requirements analysis activity

Task ID	Task List	Entry	Basic
SI.2.1	Assign Tasks to the Work Team members in accordance with their role, based on the current Project Plan.	X	X
SI.2.2	Document or update the Requirements Specification.	X	X
a	Identify and consult information sources (customers, users, previous systems, documents, etc.) in order to get new requirements.	X	X
b	Gather the identified requirements.	X	
c	Analyze the identified requirements to determine the scope and feasibility.	X	X
d	Verify the correctness and testability of the <i>Requirements Specification</i> and its consistency with the <i>Product Description</i> .	X	
e	Generate or update the <i>Requirements Specification</i> .	X	X
SI.2.3	Verify and obtain approval of the Requirements Specification. Verify the correctness and testability of the <i>Requirements Specification</i> and its consistency with the <i>Product Description</i> . Additionally, review that requirements are complete, unambiguous and not contradictory. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by analysts. If significant changes were needed, initiate a <i>Change Request</i> .		X
SI.2.4	Validate and obtain approval of the Requirements Specification.	X	X
a	Validate that the <i>Requirements Specification</i> satisfies the needs and agreed upon expectations, including the user interface usability.	X	X
b	Document the results found in SI.2.4.a in a <i>Validation Results</i> and corrections are made until the document is approved by the customer.		X
SI.2.5	Document the preliminary version of the Software User Documentation or update the present manual, if appropriate.		X
SI.2.6	Verify and obtain approval of the Software User Documentation, if appropriate. Verify consistency of the <i>Software User Documentation</i> with the <i>Requirements Specification</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by analysts. If significant changes were needed, initiate a <i>Change Request</i> .		X
SI.2.7	Incorporate the Requirements Specification, and Software User Documentation to the Software Configuration in the baseline.		X
a	Incorporate the <i>Requirements Specification</i> to the Software Configuration in the baseline.		X
b	Incorporate the <i>Software User Documentation</i> to the Software Configuration in the baseline.		X

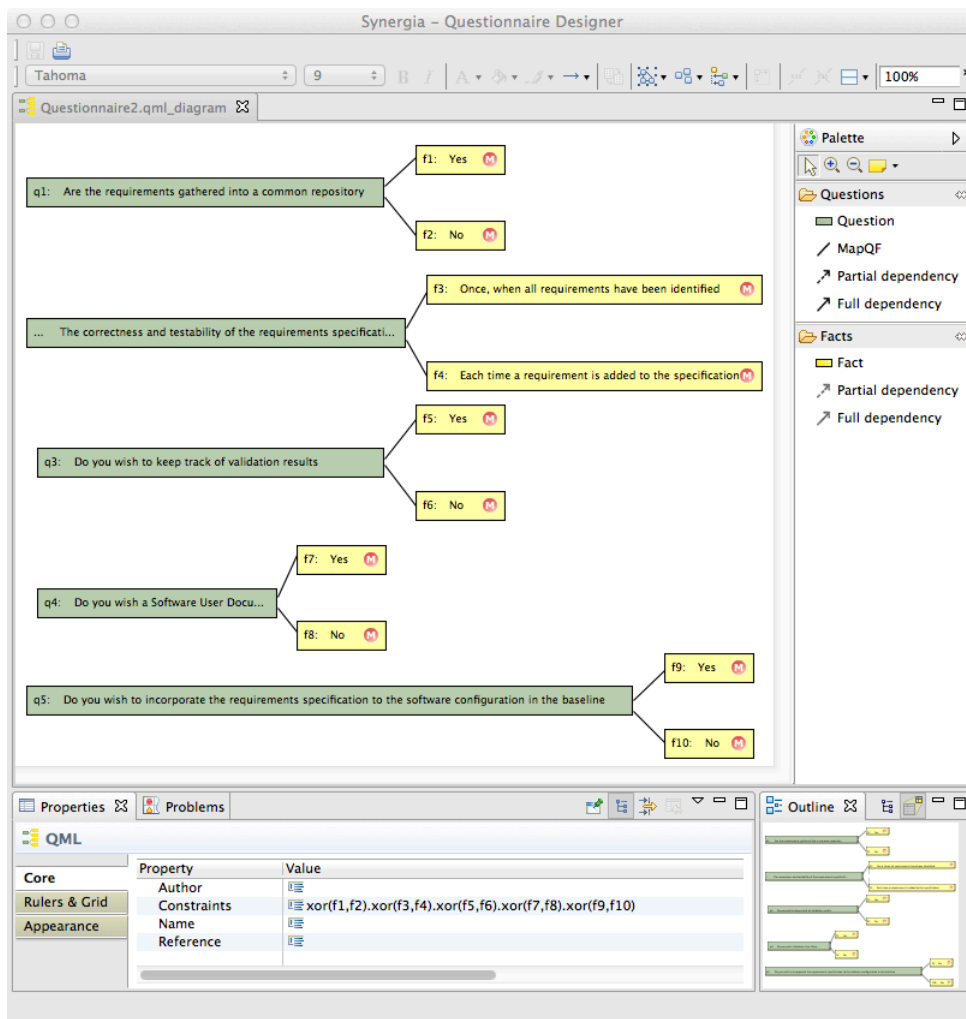


Fig. 2. Questionnaire Designer GUI

tive answer) or non selection (negative answer) of task SI.2.7.a. Finally, process individualisation is achieved directly from *Quaestio* which derives the desired YAWL process model according to the answers as described hereafter.

3.2 Deriving an Application Workflow through Individualisation

The Synergia *Quaestio* tool is the single tool used by a final user who wants to configure her workflow. Figure 3 depicts the tool's GUI. The first task of the user is to load the XML questionnaire produced by the *Questionnaire Designer*. Then, she can answer the different questions (see upper left part of Figure 3) in a random order. Upon question selection, possible answers appear in the *Question Inspector* part of the tool. Once a

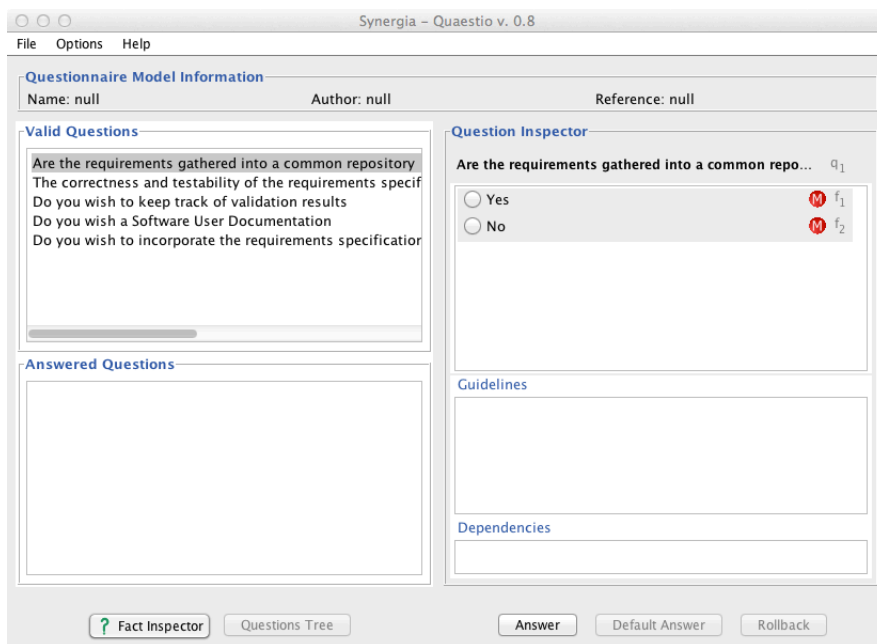


Fig. 3. Quaestio GUI

question has been answered, it will appear in the *Answered Questions* part of the tool. The user always has the opportunity to “rollback” her decisions for answered questions (*rollback* button). When all the questions have been answered, the user can individualise her process, i.e. prune the complete workflow according to her answers. For this purpose, she has to provide the tool with the paths to the workflow (.yawl) and the mapping (.cmap) previously defined. *Quaestio* will then individualise the process and save a “new” workflow containing only required tasks. A demo screencast of *Quaestio* usage is available online: <http://www.info.fundp.ac.be/~qbo/ISO29110.mov>.

4 Discussion

4.1 Lessons Learnt while Interpreting ISO/IEC 29110

Our main goal is to provide VSEs means to configure their methods according to ISO/IEC 29110. This implies that we need a deep understanding of the standard to be able to model finely a domain workflow. Even if ISO/IEC 29110 targets VSEs, its understanding still requires some basic knowledge about ISO/IEC standards which might not be available. As an illustrative example, we can mention the optionality of the *Software User Documentation* in the *Basic* profile. What is the scope of this optionality? Can the user decide to implement documentation for a given task and not others? Or is it a “global variable”? In our case, interpreting such peculiarities of the standard was not a problem as one of the authors is a member of subcommittee 7 of ISO/IEC-JTC1 defining the ISO/IEC 29110 standard. For this reason, we believe that the workflow

implemented in this paper matches the standard and its intent. However, such problems could impede the adoption of the standard by its target audience, i.e. VSEs.

Based on our modelling of ISO/IEC 29110, we actually have several points to highlight and to comment back to the national bodies contributing to ISO/IEC 29110.

First, we found that VSEs used to the *Entry* profile and wishing to upgrade to the *Basic* profile may have trouble to understand how they need to perform certain tasks when going from *Entry* to *Basic*. For instance, task *SI.2.2* of the *Entry* profile contains more subtasks than the same task of the *Basic* profile. In particular, the subtask *SI.2.2.d* of *Entry* (“Verify the correctness and testability of the *Requirements Specification* and its consistency with the *Product Description*”) is only present in the *Entry* and not in the *Basic* profile. After a more thorough analysis, the VSE may notice that the subtask *SI.2.2.d* of *Entry* is promoted to a full task in *Basic*, i.e. *SI.2.3.*. However, it is not explicitly mentioned in neither of the two profiles.

Second, given that the task order is defined through input/output relationships, it means that task numbers do not necessarily reflect task ordering. VSEs may not directly understand this fact. It would therefore be helpful not only to present workflows for the activity level but also for the task level. Furthermore, providing task workflows would help to disambiguate or verify the coherence of the proposed profile before publication. For instance, when modelling the configurable workflow for *Software Requirement Analysis*, we noticed that for the *Basic* profile, inputs to task *SI.2.6* on software documentation can already be obtained right after task *SI.2.2* since only the “Requirements Specification” is needed to conduct task *SI.2.6*. We are just wondering if this was intended or if instead, Task *SI.2.6* should require the “Requirements Specification [Validated]”.

Overall, this preliminary experience makes us confident that providing semi-automated ways to configure a process complying to ISO/IEC 29110 is a necessity to ease VSEs uptake of the standard, especially if process improvement is targeted. Indeed, understanding the differences between the profiles was the most difficult part of the work, which paramount to decide whether a given VSE will climb to the next rung.

4.2 Threats to Validity

The first threat is related to the scope of our workflow and questionnaire models: we only focused on the requirements engineering process of the SI component of the standard. Thus, there is a risk that the approach could not scale when extended to the full SI component and/or to additional profiles (*Intermediate* and *Advanced*) to come. This risk is mitigated by the fact that the configurable workflow approach/tooling on which our research relies upon has already been applied on quite complex case-studies [21].

The second threat to validity relates to the maturity of the standard itself. Indeed, we worked on a preliminary version of the *Entry* profile. Therefore, the proposed workflow may not be fully accurate and some identified variability points may evolve. Collaborating concretely with a subcommittee 7 of ISO/IEC-JTC1 member helps us to foresee future developments of the standard and the model-oriented approach to workflow configuration we have chosen eases evolution of both workflow and questionnaire models.

Finally, our approach is currently at an early stage and thus needs more development. Furthermore, it should be trialled in the field by our partners in the NAPLES

project as well as others in order to check if the proposed approach is valid. Evaluation by people familiar with the ISO/IEC 29110 standard would also be worthwhile.

5 Conclusion

In this paper, we described an approach based on configurable workflows to assist VSEs in adopting ISO/IEC 29110 compliant processes. It is implemented using the Synergia tool suite [10]. More specifically, we illustrated it with the RE process of the ISO/IEC 29110 standard. For this purpose, we first identified the variability of the RE process in the standard and represented it in a YAWL workflow. Second, we defined a user-oriented questionnaire and mapped the different answers to their corresponding tasks in the previously defined YAWL workflow. The last step is conducted by the final user who can answer those questions and the *Quaestio* tool individualises the workflow depending on her choices. The output is a YAWL workflow free of variability.

The approach is meant to lower the adoption barrier of ISO/IEC 29110 by substituting a methodology expert by a user-friendly (questionnaire-based) interface. Although, the tool might not fully replace the expert, it is likely to make ISO/IEC 29110 affordable for a wider public. The requirements workflows produced through configuration can either be followed manually (and thus used as mere documentation) or used to drive workflow which will thus enforce ISO/IEC 29110 compliance. This approach is in line with method engineering [22] techniques but is innovative in that it applies recent developments of process modelling research.

There is room for improvement. First, we would like to evaluate the approach empirically in VSEs through pilot projects that will soon start in the context of the NAPLES project. We would like to determine to what extent our approach fosters process improvement and how it can be measured. Second, we will extend the approach to new profiles as they become available as well with the project management process of the ISO/IEC standard. So far, we have assumed a sequential ordering of development tasks as an interpretation of the ISO/IEC 29110 “input/output” approach to task ordering. Although our preliminary experience shows this works for the RE process, it may be different for others. Thus, we finally would like to investigate the suitability of “declarative workflows” [23] as an alternative approach to perform ME on ISO/IEC 29110.

Acknowledgements

This work was partially funded by the Walloon Region under the NAPLES project, the IAP Programme, Belgian State, Belgian Science Policy under the MoVES project, the BNB and the FNRS.

References

1. Humphrey, W.S.: *Managing the Software Process*. Addison-Wesley (1989)
2. Institute of Electrical and Electronics Engineers, Electronics Industry Association: *Ieee/eia 12207 - industry implementation of international standard iso/iec 12207* (1998)

3. ISO/IEC: ISO/IEC 29110 – Lifecycle Profiles for Very Small Entities (VSEs) (2010)
4. Laporte, C.Y., Alexandre, S., Renault, A.: Developing international standards for very small enterprises. *IEEE Computer* **41** (2008) 98–101
5. Laporte, C.Y.: Contributions au génie logiciel et au développement et déploiement de normes internationales en génie logiciel pour de très petites organisations. PhD thesis, UBO, France (2009)
6. Saliou, P., Ribaud, V.: Iso-standardized requirements activities for very small entities. In: *Requirements Engineering in Small Companies*. (2010) 145–157
7. Alexandre, S., Laporte, C.Y.: Deployment package: Software requirements analysis - basic profile. Technical report, CETIC (Belgium) and ETS (Canada) (2007)
8. Gottschalk, F., Aalst, W.V.D., Jansen-Vullers, M.H., Rosa, M.L.: Configurable workflow models. *International Journal of Cooperative Information Systems* **17**(2) (2008) 177–221
9. La Rosa, M., Lux, J., Seidel, S., Dumas, M., ter Hofstede, A.: Questionnaire-driven configuration of reference process models. In: *Proceedings of CAiSE'07*, Springer (2007) 424–438
10. Rosa, M.L., Gottschalk, F.: Synergia - Comprehensive Tool Support for Configurable Process Models. In: *BPM (Demos)*. (2009)
11. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. *Information and Software Technology* **38**(4) (1996) 275–280
12. Henderson-Sellers, B., Ralyté, J.: Situational method engineering: state-of-the-art review. *Journal of Universal Computer Science* **16**(3) (2010) 424–478
13. Van De Weerd, I., Versendaal, J., Brinkkemper, S.: A product software knowledge infrastructure for situational capability maturation: Vision and case studies in product management. In: *Proceedings of REFSQ'06*, Springer (2006) 97–112
14. Mirandolle, D., van de Weerd, I., Brinkkemper, S.: Incremental method engineering for process improvement - a case study. In: *Engineering Methods in the Service-Oriented Context*. Volume 351 of *IFIP Advances in Information and Communication Technology*. Springer (2011) 4–18
15. Bekkers, W., van de Weerd, I., Spruit, M., Brinkkemper, S.: A framework for process improvement in software product management. In: *Proceedings of EuroSPI'10*, Springer (2010) 1–12
16. Karlsson, F., Agerfalk, P.: Method configuration: adapting to situational characteristics while creating reusable assets. *Information and Software Technology* **46**(9) (2004) 619 – 633
17. Kelly, S., Lyytinen, K., Rossi, M.: Metaedit+ a fully configurable multi-user and multi-tool case and came environment. In: *Proceedings of CAiSE'96*, Springer (1996) 1–21
18. Cervera, M., Albert, M., Torres, V., Pelechano, V.: A methodological framework and software infrastructure for the construction of software production methods. In: *Proceedings of ICSP'10*, Springer (2010) 112–125
19. ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N.: *Modern Business Process Automation - YAWL and its Support Environment*. Springer (2010)
20. Pohl, K., Böckle, G., van der Linden, F.J.: *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag New York, Inc. (2005)
21. Gottschalk, F., Wagemakers, T.A.C., Jansen-Vullers, M.H., van der Aalst, W.M.P., Rosa, M.L.: Configurable process models: Experiences from a municipality case study. In: *Proceedings of CAiSE'09*, Springer (2009) 486–500
22. Ralyté, J., Mirbel, I., Deneckère, R., eds.: *Engineering Methods in the Service Oriented Context*. IFIP W.G.8.1 Method Engineering. Volume 351. IFIP AICT - Springer (2011)
23. van der Aalst, W., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development* **23** (2009) 99–113