On the Opportunities of Scalable Modeling Technologies: An Experience Report on Wind Turbines Control Applications Development

Abel Gómez[1(✉)], Xabier Mendialdua[2], Gábor Bergmann[3], Jordi Cabot[1,4], Csaba Debreceni[3], Antonio Garmendia[5], Dimitrios S. Kolovos[6], Juan de Lara[5], and Salvador Trujillo[2]

[1] IN3, Universitat Oberta de Catalunya, Barcelona, Spain
agomezlla@uoc.edu
[2] IK4-IKERLAN Research Center, Arrasate, Spain
{xmendialdua,strujillo}@ikerlan.es
[3] MTA-BME Lendület Research Group on Cyber-Physical Systems, Budapest University of Technology and Economics, Budapest, Hungary
{bergmann,debreceni}@mit.bme.hu
[4] ICREA, Barcelona, Spain
jordi.cabot@icrea.cat
[5] Universidad Autónoma de Madrid, Madrid, Spain
{Antonio.Garmendia,Juan.deLara}@uam.es
[6] Department of Computer Science, University of York, York, UK
dimitrios.kolovos@york.ac.uk

Abstract. Scalability in modeling has many facets, including the ability to build larger models and domain specific languages (DSLs) efficiently. With the aim of tackling some of the most prominent scalability challenges in Model-based Engineering (MBE), the MONDO EU project developed the theoretical foundations and open-source implementation of a platform for scalable modeling and model management. The platform includes facilities for building large DSLs, for splitting large models into sets of smaller interrelated fragments, and enables modelers to construct and refine complex models collaboratively, among other features.

This paper reports on the improvements provided by the MONDO technologies in a software development division of IK4-IKERLAN, a Medium-sized Enterprise which in recent years has embraced the MBE paradigm. The evaluation, conducted in the Wind Turbine Control Applications development domain, shows that scalable MBE technologies give new growth opportunities to Small and Medium-sized Enterprises.

Keywords: Model-Based Engineering (MBE) · Scalability · Experience report

This work has been supported by the MONDO (EU ICT-611125) project.

A. Anjorin and H. Espinoza (Eds.): ECMFA 2017, LNCS 10376, pp. 300–315, 2017.
DOI: 10.1007/978-3-319-61482-3_18

1 Introduction

IK4-IKERLAN is a Spanish private Technology Centre focused on innovation and comprehensive product development, with 40 years of experience in combining and applying mechanics, electronics and computer science in industry. With a staff headcount of nearly 250 employees and a turnover of €18.5 million in 2015 [10], IK4-IKERLAN is a Medium-sized Enterprise [6] that has carried out advanced technology transfer for a wide variety of domains, including transportation (railway and elevators), energy (wind and solar power, and storage systems), automation, industrial, health and home appliances among others. IK4-IKERLAN has been working for the last 10 years in the development of supervisory and control platforms for wind turbines for one of the world's leading companies in the field of renewable energy.

Wind turbines, as Sect. 2 describes, are complex systems where hardware and software components need to interact in intricate ways. To tackle this complexity, Model-based Engineering (MBE) [16] technologies were introduced in 2009 in IK4-IKERLAN for the engineering of the supervisory and control systems. The goal for adopting and investing in MBE was to improve productivity and competitiveness of its industrial customers by enhancing their software development processes using, as Sect. 3 sketches, Domain Specific Languages (DSL) [7,11] and code generators [4]. The experiences reported by customers showed significant productivity increases, indicating that MBE has been critical in the development of new software products faster, cheaper and with fewer errors than previous projects.

However, too often, MBE tools and methodologies have targeted the construction and processing of small models in non-distributed environments. This focus neglects common scalability challenges [13], considering that a more typical scenario involves different engineers working in collaboration at distributed locations. Handling these issues is a challenging task that requires specific solutions that foster scalability as we will discuss in Sect. 4.

In 2013, the MONDO project was launched with the aim of tackling some of the most prominent challenges of scalability in MBE by developing the theoretical foundations and open-source implementation of a platform for scalable modeling and model management. Among the technologies developed, Sect. 5.1 focuses on the ones that can provide IK4-IKERLAN the opportunity to offer its customers software development methodologies in geographically distributed scenarios where multiple users can work collaboratively; and Sect. 5.2 describes, specifically, the different solutions developed in IK4-IKERLAN using the MONDO platform.

Section 6 describes how the scalable MONDO technologies have been evaluated in IK4-IKERLAN and the results obtained. In a scenario where the 99% of all businesses in the EU are Small and Medium-sized Enterprises (SME), Sect. 7 draws the conclusions and discusses this experience on the application of scalable modeling technologies in a company like IK4-IKERLAN.

2 Wind Turbines

A wind turbine is a complex system composed of a set of physical subsystems whose aim is to convert wind energy into electrical energy. The *Wind Turbine Control System* (WTCS) [1] is the system which monitors and controls all the subsystems that make up the wind turbine. Its aim is to maximize the generation of electrical energy, always ensuring the correct operation of the turbine and avoiding any problem which can cause any damage to it. It monitors the status of the wind turbine and the environmental conditions, making decisions to get the highest energy production. The WTCS is a HW/SW system that runs on a dedicated hardware platform. This is connected to the wind turbine through assorted communications to receive information from inputs (sensors, device state signals, etc.) and to actuate on outputs (device actuators).

The HW/SW architecture of the WTCS is shown in Fig. 1. The two lower layers refer to the HW and the Operating System. The software layer is composed of the following components:

The Execution Engine is the component which cyclically executes the algorithms to monitor and control the Wind Turbine.

The Control Units Library is a set of reusable control algorithms. These are basic blocks, with well defined interfaces, which can be instantiated and interconnected to implement the *Wind Turbine Control Application*.

A Wind Turbine Control Application (WTCA) comprises the set of algorithms that must be executed in order to ensure the correct operation of the wind turbine the WTCS is monitoring and controlling. The control algorithms of the wind turbine are specified by instantiating control units available in the *Control Units Library* and by combining those instances.

In this software architecture, the *Execution Engine* is a stable software component which does not vary from one wind turbine to another. The *Control Units Library* is also a stable component which, generally, does not vary either, unless some new device is used in a wind turbine and a custom control unit has

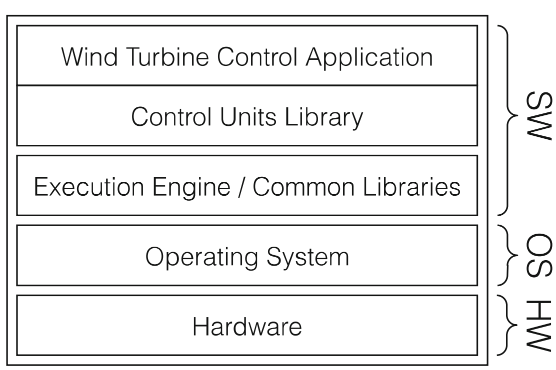

Fig. 1. HW/SW architecture of a Wind Turbine Control System

to be implemented to control it. Finally, the WTCA is the part of the software in a WTCS that is customized for each wind turbine, depending on the specific requirements that WTCS must met.

3 Model-Based Engineering for Offshore Wind Turbine Control System Development

The development of a WTCS is a process where a multidisciplinary team of hardware, software and telecommunications engineers, as well as electrical, mechanical and other engineers work in collaboration. Due to the complexity of such scenario—even considering only the software development, which comprises the three software layers described above—this paper will focus on the development of the top layer (WTCA), because the WTCA is the only part that is specific for each different wind turbine.

In recent years, IK4-IKERLAN has implemented the MBE paradigm to develop the Control Applications for Wind Turbines. This development process exploits a domain-specific modeling tool, the so-called *Wind Turbine Control Modeler* (WTCM), developed in Eclipse [19] and based on Eclipse Modeling Framework (EMF) [17]. The WTCM provides the catalogue of available *control units* that engineers can use to develop the algorithms to monitor and control the subsystems of the wind turbine.

The control system of a wind turbine is typically composed of nearly 2000 basic control units, involving nearly about 2000 inputs and up to 1000 outputs, depending on the specific model configuration. A control unit is a basic and reusable control algorithm which may be combined with other control units to build more complex algorithms. The control system is structured into logical subsystems, each controlling different physical subsystems or parts of them. The control of a wind turbine is built through the aggregation of basic control units in order to specify those complex algorithms.

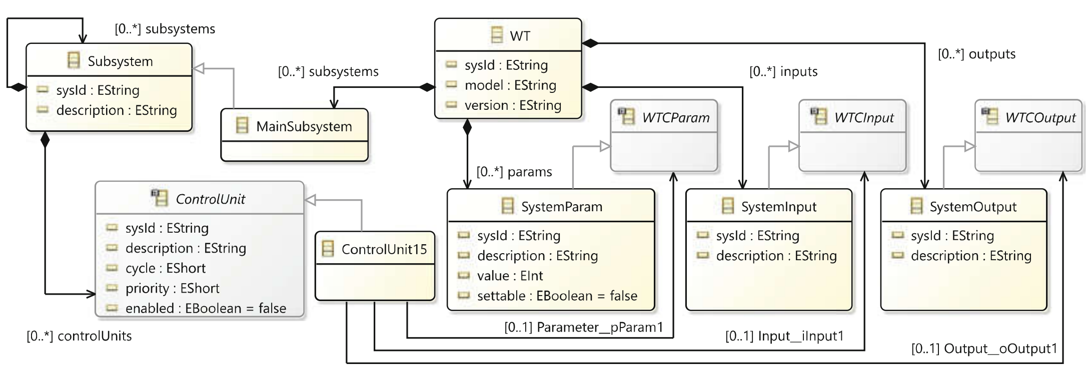

Fig. 2. Metamodel of the Wind Turbine DSL

Figure 2 shows an excerpt of the metamodel which describes the abstract syntax of the DSL provided by the WTCM. As the figure shows, a wind turbine (*WT*) contains a set of *Subsystems* which in turn contain *ControlUnits*.

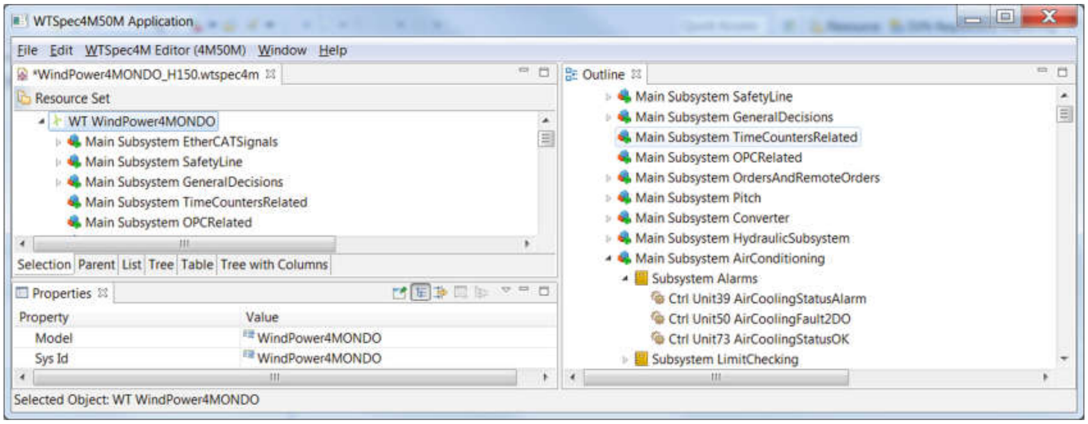

Fig. 3. Screenshot of Wind Turbine Control Modeler

ControlUnits—which may be parameterised—implement the algorithms which process a set of inputs to provide a set of outputs. For the sake of simplicity, Fig. 2 only shows the *ControlUnit15*, which processes a single input receiving a single param, and produces a single output.

Figure 3 shows what the initial implementation of the WTCM—which implements the Wind Turbine DSL—looks like. As it can be seen, the initial WTCM is an Eclipse application, which enables engineers to edit WTCS models using a regular EMF tree editor.

Once a model has been created using the WTCM, the actual C++ code for monitoring and controlling the physical subsystems can be generated using model-to-text transformations expressed in the Epsilon Generation Language [12].

4 Challenges

Today, MBE is used by a group of 10+ developers working in the R&D area of IK4-IKERLAN developing control systems for new families of wind turbines. The aim for IK4-IKERLAN is to extend MBE technologies to other activities such as wind turbine control customization for specific customer requirements. Considering that there are more than 30 different variants of control applications that are still being developed using non-MBE methodologies, it is expected that the number of different models can grow significantly within the next years, increasing the number of developers using modeling techniques up to 20 or more in the mid-term.

The main limitation, however, of the initial WTCM presented before is the lack of features that would enable a team of engineers to work collaboratively: each engineer has to work with his own copy of the model, and model merging operations—e.g., to include changes performed by others—need to be carried out manually. This manual process is a complex, tedious and error prone activity that can take more than half an hour depending on the amount and type of changes made.

Another important limitation is that engineers do not have mechanisms to work with a subset of the model. That means that all engineers must always work with the whole model, although a small subset of elements of the model can be sometimes enough to perform a specific modeling or validation activity.

Considering these limitations for the modeling solution, the following challenges have been identified to improve the development of WTCAs:

1. The **first challenge** is to move from a single-user modeling tool built for an engineer to work in an isolated way, to a modeling tool which enables several engineers to work collaboratively sharing models located in a central repository.
2. The **second challenge** is the ability to edit partial models or model fragments. Such a feature allows each engineer to work with a specific part of the model (as opposed to the whole model), thereby easing modeling activities. Additionally, the use of model fragments allows minimising the volume of data transferred over the network and limits the number of merge conflicts.
3. The **third challenge** is to graphically display and edit WTCS hierarchical models. Graphical models are more expressive for this domains as they ease the identification of relationships between model elements. This is an important enhancement with respect to the initial tree-based editor, where relationships have to be found using auxiliary views of the editor. Scalable graphical editors should include additional features like filtering facilities, hierarchies of diagrams, etc. Figure 4 shows a mockup of what a graphical WTCM would look like.
4. Finally, the **fourth challenge** is to enable model editing using a lightweight mobile device—instead of a laptop—to perform the modeling activities on site in the wind farm.

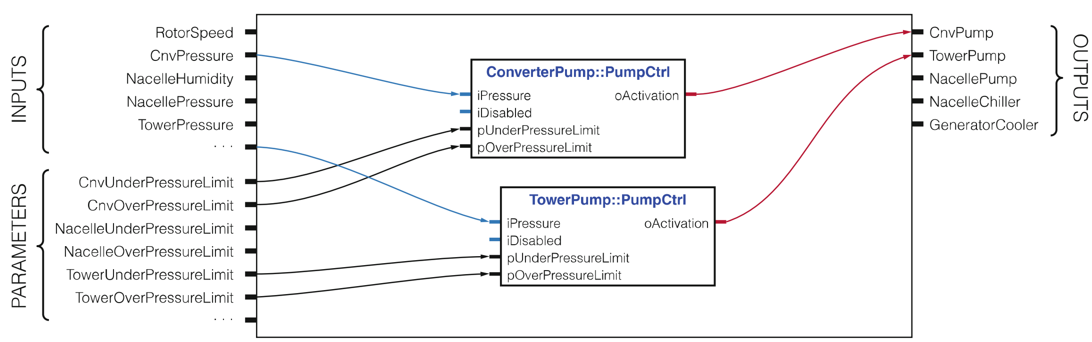

Fig. 4. Wind Turbine Control graphical modeling conceptual mock-up

5 Towards a Scalable MBE Development Process

The goal of IK4-IKERLAN joining the MONDO project was twofold: (i) to provide a real-world scenario where the scalable modeling technologies can be tested and evaluated; and (ii) to improve their development processes by introducing such techniques. In this section we first present the main MONDO technologies employed, and second, the solutions implemented in IK4-IKERLAN using these technologies.

5.1 The MONDO Platform

The MONDO platform is the open-source solution for scalable modeling and model management developed within the MONDO project. Its main purpose is to address the shortage of scalable and collaborative support in state-of-the art technologies within the MBE landscape. It is composed by several components for the development of scalable Eclipse-based editors and DSLs, collaborative modeling, model indexing and scalable model transformations and queries. The MONDO components that have been introduced in the IK4-IKERLAN development process are presented next.

The **MONDO Collaboration Framework** [3] is the MONDO component which enables collaboration both in *offline* and *online* scenarios.

In an *offline collaboration scenario*, as described in [22], models are stored in the so-called *gold repository* of a *version control system* (VCS) which is inaccessible to the actual users. VCS servers instead host separate *front repositories* dedicated to each user, that contain a copy of the gold repository, with complete version history, filtered according to the read access privileges of that user. Users are given access to a dedicated front repository so that they can read the current or historical contents of the model files (up to their read privileges) and commit their changes (which may be denied based on write permissions). In their normal day-to-day workflow, users interact with their front repository using standard VCS protocols and off-the-shelf VCS client software.

For an *online collaboration scenario*, MONDO provides an online collaborative modeling tool, where users can open, view and edit models stored in the VCS backends using a web browser, thus no client software needs to be installed. Multiple users can collaborate on the same model simultaneously, enjoying the same access control mechanism that underlies the offline collaboration framework. The editor is provided as an Eclipse RAP-based web application [20].

DSL-tao [15] is the component within the MONDO platform that enables the systematic development of scalable graphical DSLs, exploiting the idea of reusing patterns (for the domain concepts, the graphical syntax, the services of the final environment). DSL-tao profits from an extensible library of metamodeling patterns, which are instantiated and combined. The patterns account for recurring domain concepts, concrete syntax representation, semantics and services for the DSL environment. In particular, patterns have been developed to define modularization strategies for DSLs—using the **EMF-Splitter** [8] component—which result in graphical modeling environments with built-in fragmentation capabilities, and services for scoping, element visibility and filtering, among others. Using this technology, models are no longer monolithic, but split into fragments and organized similarly to programming projects. The generated environment relies on Sirius for graphical model editing, while **Hawk** [2]—the

MONDO model indexer—is used for efficient look-up of model element across fragments.

5.2 MONDO Solutions for Offshore Wind Power

Using the previous core MONDO infrastructure, three modeling solutions have been implemented in IK4-IKERLAN, which are described next.

The Online Concurrent WTCS Modeling Solution is a web modeling application which allows multiple modelers to share a modeling session. All modelers can work concurrently with the same WTCS model versioned in a model repository. All changes performed by a modeler are automatically propagated to all other modelers, having all of them an updated version of the model.

Besides allowing concurrent modeling activities, this solution also supports working with partial models. It means that each modeler can work with a different view of the same model, thus editing a different fragment of it. This is achieved by creating a custom model view containing only the model elements a user is allowed to see and edit, hiding all other elements in the model. Unauthorized editing of model elements is also prevented. The *Online Concurrent WTCS Modeling solution* allows modelers to commit the changes performed in the model to the model repository.

The Offline Collaborative WTCS Modeling Solution is an Eclipse application which runs locally. It enables several engineers to work with a shared model, but unlike the solution presented above, collaborative modeling is done asynchronously, i.e., each modeler working with the shared model edits a local copy of the shared WTCS model, which is checked out (or updated) from a model repository. When model editing has finished—or whenever the user decides—a commit operation is requested and the MONDO Collaboration Framework carries out the operation.

This asynchronous way of working with a WTCS model may lead to conflicts when several engineers make changes to the model and try to commit them. However, to avoid this, the MONDO Collaboration Framework will assist the modeler to resolve the conflicts by merging the remote and local changes consistently using an automated search-based model merge [5] before the commit succeeds.

As aforementioned, in MONDO, the management of partial models in an offline manner is handled by the MONDO Collaboration Framework which performs the synchronization between all the front repositories and the gold repository. In the case of the *Offline Collaborative WTCS Modeling solution*, there exists a different front repository for each different user type which contains the partial model for that user type.

Apart from collaboration related operations, the way a model user will work with the *Offline Collaborative WTCS Modeling solution* is quite similar to the way the modeler was working with the tree-based single user modeling tool introduced in Sect. 3.

The Offline Graphical Collaborative WTCS Modeling Solution is a Sirius-based editor which allows editing WTCS models graphically. As mentioned above, WTCS models edited by the offline modeling solution are built with the Wind Power domain specific tree editor, but, as mentioned in Sect. 4, an important challenge is the capability of editing WTCS models graphically, with an editor which provides advanced features like drill-down, element filtering, layers, custom views, different diagrams, etc.

This Offline Graphical Collaborative editor addresses this challenge by exploiting DSL-tao. DSL-tao was used to define the graphical syntax by instantiating the pattern for concrete syntax using its dedicated wizard, and to generate the Sirius-based editor. It is noteworthy that a wind turbine model can be edited either using the classical EMF-based tree editor or the graphical editor based on Sirius.

6 Evaluation

In order to assess the success of the MONDO technologies in the IK4-IKERLAN MBE processes, an evaluation has been performed. This evaluation—whose details are extensively covered in the project deliverables [22]—is reported next.

6.1 Evaluation Framework

Three realistic scenarios, which express the four challenges presented in Sect. 4, are used for the validation of the MONDO technologies:

Scenario S1: Wind turbine control design — Three system engineers work concurrently on a single model, modeling different subsystems. Each system engineer works on a partial model or submodel and MONDO technologies shall merge all the partial models into a unified model.

Scenario S2: Wind turbine commissioning — A system engineer works on a partial submodel of a model during the commissioning of a subsystem. Transformations for code generation will only take into account the artefacts contained in (and referenced from) the submodel the engineer is working on.

Scenario S3: Maintenance activities in the wind farm using mobile devices — A maintenance operator of a wind farm detects a malfunction of a non-critical element in a wind turbine. This causes the wind turbine to be out of operation. The engineer makes a minor change in the control model and obtains new code to put the wind turbine into operation in a degraded mode. These minor changes are made using a tablet or a mobile device.

The solutions presented in Sect. 5.2 support the three aforementioned scenarios:

Scenario S1 can be supported by two MONDO solutions: the *Online Concurrent WTCS Modeling Solution* and the *Offline Collaborative WTCS Modeling Solution*. Although in a different way, both solutions allow several engineers to work in parallel on the algorithms for the different subsystems of a wind turbine. Likewise, both solutions manage all changes performed by each engineer merging them in a single WTCS model.

Scenario S2 is supported by the *Offline Collaborative WTCS Modeling Solution*. A specific subsystem manager is allowed to load only a fragment of the entire WTCS model. Thus she can only edit the part of the model for the subsystem under her responsibility. The subsystem manager also has the ability to generate code for the subsystem.

Scenario S3 is supported by the *Online Concurrent WTCS Modeling Solution*. As this is a web based solution deployed on a web server, it can be accessed using a tablet. Thus, on-site modeling operations related to maintenance activities can be performed by the maintenance operator.

In order to better measure the impact of the MONDO technologies in the development of a WTCA, the following top-level indicators have been identified: (i) *time for committing model changes*; (ii) *impact on performance derived from using MONDO Collaboration technology*; and (iii) *time reduction for building graphical domain specific modeling editors*. These generic indicators have been materialized into a set of quantitative and qualitative measures, which are summarized in Tables 1 and 2. Based on the expertise of IK4-IKERLAN in the domain, different criteria have been defined to measure the success of the quantitative measures. On the other hand, the qualitative evaluation is carried out amongst the engineers participating in the evaluations using the questions in Table 2. In their answers, a four point scale (i.e., *fully*, *largely*, *partially*, *none*) is used together with the opportunity for respondents to provide comments and clarifications regarding their assessment.

Table 1. Quantitative measures and evaluation criteria

Id	Description	Sufficient	Good	Excellent
QN1	Increase in time for loading a model on a tablet instead of on a PC	25%	15%	10%
QN2	Number of concurrent users working with a model	2	3	5+
QN3	Time for change propagation and notification among concurrent users	<5 s	<3 s	<1 s
QN4	Maximum number of elements that can be displayed in a diagram	25	50	>50
QN5	Time for loading a diagram having 25 elements to be displayed	2 s	1 s	<1 s
QN6	Time for committing model changes	<5 s	<3 s	<1 s
QN7	Performance impact caused by the MONDO Collaboration Framework	<5%	<2%	<1%
QN8	Time reduction for building graphical domain specific modeling editors	25%	50%	75%

Table 2. Qualitative measures

Id	Description
QL1	Is there a methodology which specifies how a large DSL should be constructed?
QL2	Is there a tool support for the methodology, which guides the user on the construction of a large DSL?
QL3	Does this tool provide a way to create a basic but fully functional collaborative domain specific modeling tool?
QL4	Is MONDO technology mature enough to be used in industrial solutions?
QL5	Does MONDO technology allow concurrent editing of a model?
QL6	Does MONDO technology allow partial loading of models?
QL7	Does MONDO technology allow progressive loading of a model?
QL8	Does MONDO technology allow working with several modeling languages in a single tool?
QL9	Can a model be edited using a tablet?

6.2 Evaluation Results

In order to perform the evaluation, a test environment was set up as described in [22]. This environment consisted of a group of 6 different types of domain engineers—with different privileges—working together in collaborative modeling tasks using state-of-the-art tablets, desktop and laptop PCs.

Quantitative Measures

QN1 — The *Online Concurrent WTCS Modeling solution* has been used to evaluate this measure. Analyzing the data collected, we observe that the increase of time for loading the model on a tablet instead of on a PC, is between 11% and 13% for model fragments and goes up to 20% when full model is loaded.

QN2 — This measure has been evaluated using the *Online Concurrent WTCS Modeling solution*. The modeling solution has been successfully executed—i.e., without performance loss—by five different users on the same shared modeling session.

QN3 — Time needed to propagate changes and notifying them among concurrent users has been evaluated using the *Online Concurrent WTCS Modeling solution*. The solution has been executed by five different users, four of them were working with a desktop PC, while fifth one worked on a tablet. Each time a modification was made by one user, the time required for notifying other users has been measured. Less than one second was needed to propagate the modifications with a standard network equipment.

QN4 — The *Offline WTCS Graphical Modeling solution* has been used to evaluate this measure. In this evaluation, several diagrams have been loaded in this editor with scalable capabilities such as drill-down and filtering. Their sizes ranged from 5–15 elements to nearly 450 elements. All the diagrams were successfully loaded.

QN5 — The *Offline WTCS Graphical Modeling solution* has been used to evaluate this measure. The measurements show that loading and displaying a diagram with 25 elements takes between two and three seconds. However, although the target measure is not met, the result achieved is very close to the target result. It is also noteworthy that the largest diagram (with 452 elements) can be loaded in less than 7 s. As a consequence, although the target result has not been met, the overall results achieved in this evaluation are considered sufficient.

QN6 — This measure has been evaluated with the *Offline Collaborative WTCS Modeling solution*. When a commit operation is performed in an offline scenario, model changes are committed to the gold repository and propagated to the front repositories for the different engineer types who are working in collaboration.

The first conclusion after the evaluation is that the number of changes to be committed to the repository has no significant impact on the time needed to commit them. Specifically, the times collected for committing changes and updating the front repositories range from two to three seconds for every front repository, regardless of the number of changes. It is important to highlight that model merging was a hand-made and error prone activity which could take up to half an hour before the introduction of the MONDO technologies [5].

QN7 — Impact on performance has been evaluated using the *Offline Collaborative WTCS Modeling solution*. The solution designed and developed for offline collaboration does not penalize the performance of features that were already available in the solution. In this sense, an engineer gets the same performance for the modeling activities she was used to carry out, but with the addition of the collaborative modeling capabilities.

QN8 — The process of construction of the *Offline WTCS Graphical Modeling solution* has been considered for the evaluation of this measure. The aim of this measure is to compare the time required to create a graphical wind turbine control system modeling editor using MONDO technology (concretely DSL-tao and its design patterns), and time required to create the graphical modeling editor using the design tool provided by Sirius. Specifically, a graphical WTCS modeling editor prototype can be constructed using the Sirius graphical specification in nearly two hours by an expert developer on this technology. The equivalent graphical modeling editor has been constructed using DSL-tao in no more than half an hour by the same developer with basic—but sufficient—knowledge of DSL-tao.

Table 3 summarizes the rates achieved in the different quantitative measures according to the criteria specified in Table 1.

Table 3. Quantitative measures Results

Id	Rating	Id	Rating	Id	Rating	Id	Rating
QN1	Sufficient	QN3	Excellent	QN5	Sufficient	QN7	Excellent
QN2	Excellent	QN4	Excellent	QN6	Good	QN8	Excellent

Qualitative Measures. Besides the quantitative measures presented above, a set of qualitative measures were also planned. The fulfillment of this set of measures is explained below, in Table 4, where the level of compliance for each measure is presented along with some additional comments. The level of compliance is set using a four point scale with the following values: (i) *fully*, the expected target measure has been achieved for the Wind Power domain; (ii) *largely*, although the target measure has not been fully been reached, the achieved result is very close to the expected result; (iii) *partially*, some interesting results have been achieved for the Wind Power domain thanks to MONDO technology, although the target measure has not been reached; (i) *none*, no result related to the target measure has been obtained with the MONDO technology.

Table 4. Qualitative measures results

Id	Fulfillment	Comments
QL1	Fully	DSL-* tools provide a step by step process for designing large DSLs.
QL2	Fully	The tool supporting large DSL construction is DSL-tao. It provides a set of design patterns to design the DSL and to build its modeling tool
QL3	Largely	A functional domain specific modeling tool can be created using DSL-tao, but collaboration features are not fully supported
QL4	Largely	Components like the MONDO Collaboration Framework are ready to use. Setting up of the collaboration environments, however, should be automated
QL5	Fully	This feature is provided by MONDO Online Collaboration Framework, which has been used to build the *Online Concurrent WTCS Modeling solution*
QL6	Fully	This feature is provided by the MONDO Collaboration Framework. EMF-Splitter provides also this feature, enabling to split a model into different physical files that can be loaded separately on demand
QL7	Fully	Progressive loading can be achieved by EMF-Splitter where each model fragment can be loaded on demand, when modularity pattern is applied
QL8	Fully	Although this requirement has not been validated in the previous use cases, the MONDO technology has been tested to confirm that there is no constraint to combine two different modelling languages in a single modelling tool
QL9	Fully	The MONDO Collaboration Framework allows users to edit a model concurrently using a web modeling application run on a tablet

Table 5. Scenarios, solutions and evaluated measures

Scenario	MONDO solution	Measure id
S1	Online concurrent	QN2, QN3, QL5, QL6, QL4
S1	Offline collaborative	QN4, QN5, QN6, QN7, QN8, QL1, QL2, QL3, QL4, QL6, QL7, QL8
S2	Offline collaborative	QN6, QN7, QL1, QL2, QL3, QL4, QL6
S3	Online concurrent	QN1, QL4, QL9

Scenario Coverage. Table 5 summarises the relationship among the evaluation scenarios described in Sect. 6.1, the MONDO solutions used for each scenario, and the measures evaluated by the solution in each scenario.

7 Discussion

Scalable modeling technologies can provide new opportunities to SME to grow their software development teams. In this document we have reported on the experience at IK4-IKERLAN after implementing the technologies developed in the MONDO project. The experience has been extremely positive, and the evaluation shows that five out of eight quantitative measures scored excellent—one of them scored good and two others scored sufficient—while seven out of nine qualitative measures were fully fulfilled—the two remaining were largely fulfilled.

From this experience, we can also learn that **continuous compliance** with existing development processes is a key factor for success. The MONDO scalable technologies do not impose a big change on the processes and tools that were already implemented in the company. In this sense, the new solutions enable teamwork in the offline scenario in such a way that can be integrated without changing the pre-existing single-user modeling tools. This way developers continue working in the same way they used to work, and collaboration features only come into play to automate operations that were manual before (e.g., model merging).

Part of this success has been due to, not only the technology itself, but to the **methodological guidance** provided by MONDO. Specifically, the methodology supported by DSL-tao can be easily followed to construct large scale DSLs. In this sense, it is important that this methodology provides a wide set of predefined design patterns, which DSL designers can take advantage of to build their custom modeling solutions.

Another important contribution of the scalable technology is the capability for **concurrent model editing using web technology**, enabling real-time collaboration with secure access control, even using mobile devices. While there are several emerging modeling frameworks to support web-based collaborative modeling such as AToMPM [18], WebGME [14], Web Modeling Framework [21]—see [9] for an overview—security and scalability remains a major challenge for them.

As demonstrated by *Online Graphical Collaborative WTCS Modeling Solution* the Eclipse RAP platform [20] is not mature enough.

Finally, this experience also evidences that web-based solutions are not best suited to carry out modeling activities in handheld mobile devices, since they present usability issues. In this sense, another possible avenue for research is the development of **dedicated domain-specific modeling environments for mobile devices** [23].

Acknowledgements. We would like to thank István Ráth, Dániel Varró, and all the MONDO researchers for their contributions to the project.

References

1. Ackermann, T., Söder, L.: Wind energy technology and current status: a review. Renew. Sustain. Energy Rev. **4**(4), 315–374 (2000). doi:10.1016/S1364-0321(00)00004-6
2. Barmpis, K., Kolovos, D.: Hawk: towards a scalable model indexing architecture. In: Proceedings of the Workshop on Scalability in Model Driven Engineering, pp. 6:1–6:9, BigMDE 2013, NY, USA (2013). doi:10.1145/2487766.2487771
3. Bergmann, G., Debreceni, C., Ráth, I., Varró, D.: Query-based access control for secure collaborative modeling using bidirectional transformations. In: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, Saint-Malo, France, 2–7 October 2016, pp. 351–361 (2016).doi:10.1145/2976767.2976793
4. Czarnecki, K., Eisenecker, U.W.: Generative Programming: Methods, Tools, and Applications. ACM Press/Addison-Wesley Publishing Co., New York (2000). ISBN:0-201-30977-7
5. Debreceni, C., Ráth, I., Varró, D., Carlos, X., Mendialdua, X., Trujillo, S.: Automated model merge by design space exploration. In: Stevens, P., Wąsowski, A. (eds.) FASE 2016. LNCS, vol. 9633, pp. 104–121. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49665-7_7
6. European Commission: What is an SME? http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_es. Accessed Feb 2017
7. Fowler, M.: Domain Specific Languages, 1st edn. Addison-Wesley Professional, Upper Saddle River (2010)
8. Garmendia, A., Guerra, E., Kolovos, D.S., de Lara, J.: EMF splitter: a structured approach to EMF modularity. In: Proceedings of XM@MODELS. CEUR Workshop Proceedings, vol. 1239, pp. 22–31. CEUR-WS.org (2014). http://ceur-ws.org/Vol-1239/xm14_submission_3.pdf
9. Gray, J., Rumpe, B.: The evolution of model editors: browser- and cloud-based solutions. Softw. Syst. Model. **15**(2), 303–305 (2016). doi:10.1007/s10270-016-0524-2
10. IK4-IKERLAN: Efficiency in service innovation for companies—IK4-IKERLAN. http://www.ikerlan.es/en/ikerlan/. Accessed Feb 2017
11. Kleppe, A.: Software Language Engineering: Creating Domain-Specific Languages Using Metamodels, 1st edn. Addison-Wesley Professional, Upper Saddle River (2008). ISBN:0321553454, 9780321553454
12. Kolovos, D.S., Paige, R.F., Polack, F.A.C.: The epsilon transformation language. In: Vallecillo, A., Gray, J., Pierantonio, A. (eds.) ICMT 2008. LNCS, vol. 5063, pp. 46–60. Springer, Heidelberg (2008). doi:10.1007/978-3-540-69927-9_4

13. Kolovos, D.S., Rose, L.M., Matragkas, N., Paige, R.F., Guerra, E., Cuadrado, J.S., De Lara, J., Ráth, I., Varró, D., Tisi, M., Cabot, J.: A research roadmap towards achieving scalability in model driven engineering. In: Proceedings of the Workshop on Scalability in Model Driven Engineering, pp. 2:1–2:10. BigMDE 2013, NY, USA (2013).doi:10.1145/2487766.2487768
14. Maróti, M., Kecskés, T., Kereskényi, R., Broll, B., Völgyesi, P., Jurácz, L., Levendovszky, T., Lédeczi, Á.: Next generation (meta)modeling: web- and cloud-based collaborative tool infrastructure. In: Proceedings of the 8th Workshop on Multi-Paradigm Modeling (MPM) co-located with MODELS 2014, pp. 41–60 (2014). http://ceur-ws.org/Vol-1237/paper5.pdf
15. Pescador, A., Garmendia, A., Guerra, E., Cuadrado, J.S., de Lara, J.: Pattern-based development of domain-specific modelling languages. In: 18th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MoDELS, pp. 166–175. IEEE Computer Society (2015). doi:10.1109/MODELS.2015.7338247
16. Selic, B.: The pragmatics of model-driven development. IEEE Softw. **20**(5), 19–25 (2003). doi:10.1109/MS.2003.1231146
17. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework 2.0, 2nd edn. Addison-Wesley Professional, Amsterdam (2009). ISBN:0321331885
18. Syriani, E., Vangheluwe, H., Mannadiar, R., Hansen, C., Mierlo, S.V., Ergin, H.: AToMPM: a web-based modeling environment. In: Joint Proceedings of MODELS 2013 Invited Talks, Demonstration Session, Poster Session, and ACM Student Research Competition, pp. 21–25 (2013). http://ceur-ws.org/Vol-1115/demo4.pdf
19. The Eclipse Foundation: Eclipse - The Eclipse Foundation open source community website. https://eclipse.org/. Accessed Feb 2017
20. The Eclipse Foundation: Remote Application Platform (RAP). http://eclipse.org/rap/. Accessed Feb 2017
21. The Eclipse Foundation: Web Modeling Framework (previously genmymodel.com). https://projects.eclipse.org/proposals/web-modeling-framework/. Accessed Feb 2017
22. The MONDO Project: Work Package 4 - Scalable Collaborative Modelling. Deliverable 4.4: Prototype Tool for Collaborative Modeling. http://hdl.handle.net/20.500.12004/1/P/MONDO/D4.4
23. Vaquero-Melchor, D., Garmendia, A., Guerra, E., de Lara, J.: Towards enabling mobile domain-specific modelling. In: Proceedings of the 11th International Joint Conference on Software Technologies - Volume 2: ICSOFT-PT, (ICSOFT 2016), pp. 117–122 (2016).doi:10.5220/0006002501170122