

Tooling: Improved Management for Monitor Repositories and Measuring Points in Palladio

Florian Nieuwenhuizen, Domas Mikalkinas, Lasse Merz, David Schütz,
Ba-Anh Vu, Birasanth Pushpanathan, Mehmet Ali Tepeli

{st111292 | st113254 | st110035 | st110033 | st100483 | st154231 | st154110}@stud.uni-stuttgart.de
University of Stuttgart, Stuttgart, GER

Abstract

The SimuLizar extension to the Palladio Component Model brought new features to the simulator. One of these features allows users to manually define Monitors and Measuring Points (M&Ms). Leveraging M&Ms users can reduce simulation memory demand and increase its efficiency while offering the capabilities to specify exactly what needs to be analyzed.

However, the current implementation makes it very difficult for new users to create and understand M&Ms. Therefore, we aim to provide an easier-to-use UI, through an Eclipse wizard, increasing the usability by guiding users through the set-up process in an easy and understandable way.

These additions are extensible, allowing future developers to extend the functionality of our solution through Eclipse extension points.

1 Introduction

With its approach to software architecture modeling and simulation, Palladio [2] became a widely used simulator, not only in research but also by companies. Naturally, over time the software was subject to many changes and additions, due to the need for maintenance and adaptation to new technological achievements in the field of software development.

One of the new contributions to the Palladio project is SimuLizar [3]. SimuLizar can be used to simulate self-adaptive systems, especially such using the fast-growing cloud computing paradigm. To analyze such systems, SimuLizar introduces new metrics to the PCM: Scalability, elasticity and efficiency. This is achieved by using Monitors, Measuring Points and multiple measurable metrics. Monitors are used for specifying which metrics should be measured while Measuring Points determine the models which are measured. This allows the users to measure only those parts of the model they are interested in, rather than measuring the whole system.

While the PCM user interface is well-developed and mature, the user interface of SimuLizar utilizes basic EMF forms for presenting its models. This results in the problem, that in the current version of SimuLizar, the provided user interface for managing

Monitors and Measuring Points is very complex and unintuitive. This leads to frustration of the user or to failures while executing the simulation.

We asked ourselves how we could improve the Measurement creation process for the user, making it easier and more understandable how to create Measurements, as well as removing invalid combinations that can result in failures.

Our attempt of improving the usability in the M&M creation process provides a usable UI to help users simulate their projects properly. This is achieved by providing multiple possibilities to manage Monitors and Measuring Points, through an Eclipse view and a wizard.

Since SimuLizar is a changing and extendable project, our approach also has to meet these requirements. More precisely, SimuLizar can be extended by additional types of Measuring Points or metrics. Therefore our approach has to be adaptable in order to give the user valid recommendations and to prevent failures during the Monitor and Measuring Point creation process.

In this paper, we explain the current state of the Measurements creation process, introduce and describe our approach, as well as further features that could be implemented in the future.

2 Current State of Measurements

As mentioned, in SimuLizar the user has to specify Monitors and Measuring Points for specific models to be able to perform simulations. The current workflow to create them is explained in the following: After the user modeled his system including all necessary Palladio component models, he can start creating M&Ms. First, the user has to create a repository for Monitors as well as for Measuring Points. These repositories are used to store and manage the corresponding objects. There exists a simple UI for each repository.

He starts with the Measuring Point repository, where he creates a new Measuring Point. The user has to select one of 14 different types of Measuring Points. They each correspond to a model element of the Palladio component model. Some of the Measuring Points are self-explanatory, while others are un-

clear in what they correspond to. After he created a Measuring Point he has to assign the corresponding model element to it. To accomplish this task the user has to drag-and-drop the model which contains the element into the UI of the Measuring Point repository and assign it in the properties of the Measuring Point.

After switching over to the Monitor repository, he has to create a new Monitor. To connect the created Measuring Point with this Monitor he has to load the Measuring Point repository via drag-and-drop into the Monitor repository. Only after loading it, the Measuring Point can be linked to the newly created Monitor. The Monitor specifies which metrics should be used for Measurements during a simulation run. Currently, there are over 60 metrics to choose from. It is possible to add all of them to the Monitor, but only a few can actually be used to measure a specific Measuring Point. Also, there is no documentation or help from the software to determine the correct metric.

Additionally, the user can specify how the Measurements of each metric are processed. In the default option they are passed directly to the recorder. There are also options to aggregate the Measurements in some way before passing them.

If the user executed all these steps correctly and he chooses only metrics that can be used for his Measuring Point, the simulation will run through and he can view the measured results. Unless the user is experienced, this probably won't be the case though and there are reasons for this.

Not only are there many steps to perform, but the UI to achieve this is also very inconvenient and incomprehensible. Unfortunately, there is no support within the software, which explains any of the steps or choices he can make. This can prevent new users from using the software at all, but also often leads to failing execution of the Measurement.

To solve this problem, we designed and developed a new, user-friendly way of creating Monitors and Measuring Points with usable metrics.

3 Wizard-based Concept

Our approach to help users create Monitors and Measuring Points combines a dashboard-style overview with comprehensive wizards that guide the user through the entire process.

The main goals of our approach consist of the following:

- Guide the user through the entire process and thereby eliminate the potential confusion of which step to take next.
- Eliminate the clutter of different options that occur during the process.
- Present only valid options in the respective context and thereby ensure a correct and working result.

- Give suggestions and explanations for novice users.
- Provide advanced options for experienced users.

To reach these goals, we incorporated an overview which we refer to as the 'dashboard'. Like in a real dashboard, this view contains an overview showing the current state of the system and its parameters. In our case, the user has access to all Monitors and Measuring Points that currently exist in the system. These are presented in a tree structure, indicating the hierarchy and containment of each element (see Figure 1(a)). All important aspects and information are bundled in one location and in this way the user can gain a quick overview of the current state.

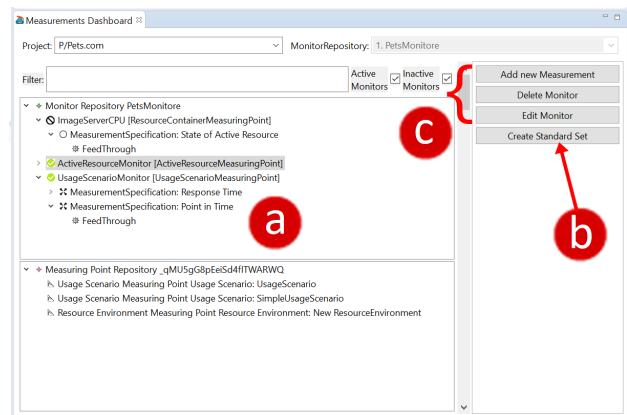


Figure 1: Our novel 'dashboard' view, which gives an overview over the current state of all M&Ms and also grants access to other related functions.

In addition to this, the dashboard provides quick access to various functions which are related to the creation and manipulation of Measurements for this project. The 'Standard Set' function automatically creates Monitors and Measuring Points depending on existing models (see Figure 1(b)).

Other functions regarding the manipulation of M&Ms (see Figure 1(c)) open their own respective wizards which guide the user through the according process, much like other existing wizards within Eclipse. The wizards are designed to fit seamlessly in Eclipse, so there is no adaption needed.

During the use of the wizard, the users are guided through a step-by-step process where they can choose between different options and define parameters for according settings (see Figure 2). To avoid frustration for the user, it is ensured that only options and settings are enabled that lead to a working and correct result. After the user created a Monitor in the first step, we provide the user with a list of model elements from his project in the second step. Selectable elements are marked bold and the corresponding Measuring Point is created automatically. In the third step, we only show metrics that can be used with the

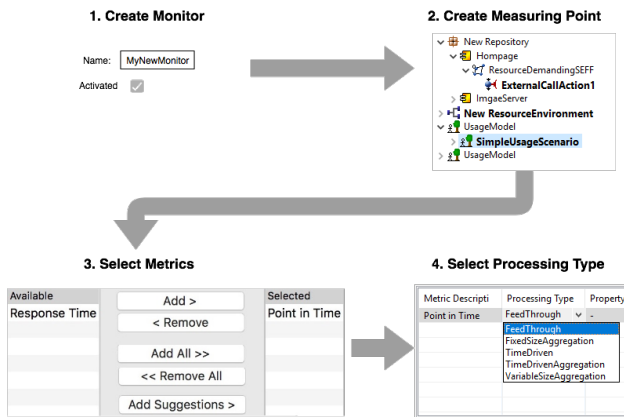


Figure 2: The workflow of our wizard to create a Monitor with a Measuring Point and usable metrics.

selected Measuring Point. Optionally, we provide suggestions for common options that the user can choose from. This is especially useful for novice users that are new to Palladio and otherwise would be unsure of which ones to choose. We designed this part to be extendable through an extension point, where developers define usable metrics and suggestions. While novice users can finish after this step, we provide experienced users with the option to define a processing type for each metric in the fourth step.

At the end of each wizard, the specified repositories and objects (Monitors, Measuring Points etc.) are automatically created and assigned to their respective models within the system. This again represents a speed-up for the work-flow of the user, as otherwise, the steps would have to be performed manually.

Each wizard page provides usable UIs. These pages have been created with EMF Parsley [1]. EMF Parsley provides reusable UI components based on EMF models (trees, tables, forms) for Eclipse plugin projects. EMF model parts can be easily adapted with the domain specific language Xtext. Using EMF Parsley, user-interface components of EMF models can be developed conveniently. Since the PCM uses EMF, our solution can be built on the existing system using EMF Parsley.

4 Future Work

Besides creating and editing Monitors and Measuring Points through a wizard, there are other alternative ways to do this task. Similar to electrical circuits, a Measuring Point can be inserted exactly where it will be measured. For such an approach the already provided Sirius [4] graphical editors of the PCM can be extended. Thus the Measuring Point can be set to the place in the PCM, where the Measurement is to be made. Important, however, would be that both the wizard approach as well as the Sirius approach can be used independently of each other.

In addition, our approach could be applied to ex-

periment automation, another element of SimuLizar which doesn't have a usable UI yet.

5 Conclusion

The addition of SimuLizar to the Palladio Component Model offered users the possibilities to simulate self-adaptive systems. SimuLizar introduced new components for measuring these systems. To create a new simulation, the users need to create Monitors and Measuring Points (M&Ms) in combination with some of many metrics. As the current implementation does not focus much on usability, many users, especially users without further knowledge about SimuLizar, are not capable of creating M&Ms with correct metrics. As there are over 60 different, unexplained metrics to choose from, with no information about which metrics result in valid simulations, we decided to create a different approach to aid users in creating valid simulations.

Our approach combines a dashboard-style overview with a wizard that guides the user through the entire process of creating M&Ms, step-by-step. It eliminates options leading to invalid simulations, only presenting valid options, depending on the context of the model. Furthermore, the wizard offers suggestions for novice users, while providing further options for experienced users.

Since the SimuLizar extension is an ongoing project with new Measuring Points and metrics still being added, we offer extensibility through Eclipse extension points, allowing developers to integrate their additions into the UI of our solution.

Future developers could attempt to add a Sirius-based approach, allowing the creation and modification of M&Ms directly in the graphical editors of the PCM.

References

- [1] *Eclipse Modeling Project EMF Parsley*. <https://projects.eclipse.org/projects/modeling.emf-parsley>. Accessed: 2018-08-31.
- [2] *Palladio The Palladio Component Model*. <https://www.palladio-simulator.com/home/>. Accessed: 2018-10-07.
- [3] *SimuLizar Palladio Plug-in*. <https://sdqweb.ipd.kit.edu/wiki/SimuLizar>. Accessed: 2018-10-07.
- [4] *Sirius The easiest way to get your own Modeling Tool*. <https://www.eclipse.org/sirius/>. Accessed: 2018-08-31.