Runtime Adaptations within the QuaD^2-Framework

Steffen Mencke, Martin Kunz and Mario Pukall

Otto-von-Guericke University, P.O. Box 4120, 39016 Magdeburg, Germany

Abstract The importance of providing integration architectures in every field of application is beyond controversy these days. Unfortunately, existing solutions are focusing mainly on functionality. But for the success of Systems Integration in the long run, the quality of developed architectures is of substantial interest. Therefore, a framework for quality-driven creation of architectures is proposed in [1]. The idea fundamentally bases on functional and non-functional runtime adaptations.

1 Introduction

Due to manifold advantages of high-flexible infrastructures compared to monolithic products a lot of initiatives propose approaches for the integration of single components (e.g. services, content). Semantic metadata provides the basis for the automation of this process. But those approaches lack from a throughout consideration of empirical data. Either only functional requirements or single quality attributes are taken into consideration.

The presented general QuaD^2-Framework (Quality Driven Design) is intentionally described in an abstract way to enable an applicability to different fields, e.g. e-learning content provision, service oriented architectures and enterprise application integration. For this reason a general terminology is used and special domain-specific instantiations are described elsewhere (e.g. in [2] or [3]).

In contrast to existing approaches the QuaD^2-Framework reveals a holistic orientation on quality aspects. It combines semantic web technologies for the fast and correct assembly of elements and quality attribute evaluations for the best possible assembly decisions.

Several points of runtime adaptations reveal the advantages of the framework in order to enable an up-to-date entity assembly and presentation. In fact that targets the quality-driven selection of appropriate entities as well as the experience-based selection of process and quality models.

2 QuaD^2-Framework

The major goal of the described core process is the assembly of an infrastructure consisting of single entities. Such an entity is metadata-annotated functionality and may be depicted by e.g. services, agents or content fragments in concrete applications.

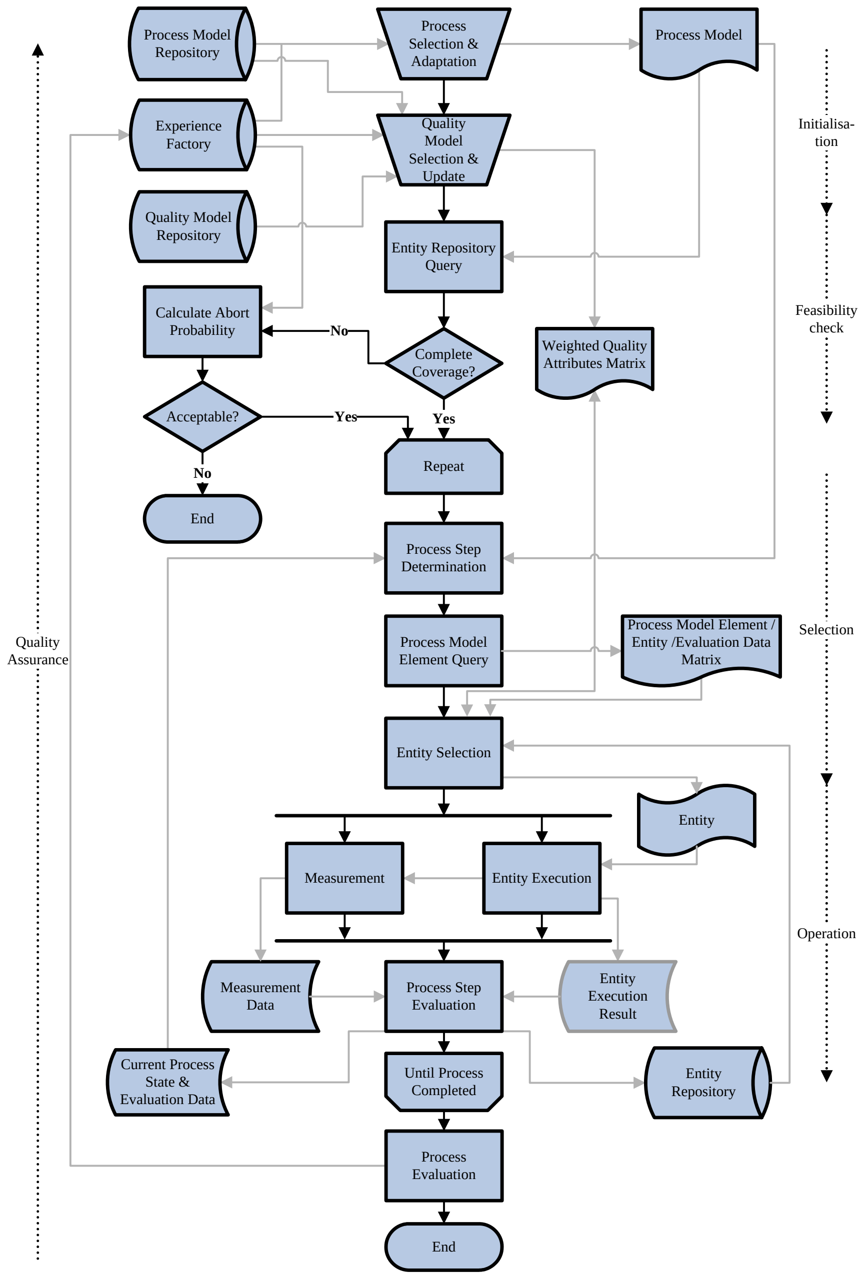

Figure 1. QuaD^2-Framework [1].

The QuaD^2-Framework is shown in Figure 1.

In general the subprocesses of this empirical-based assembly process are the initialization, the feasibility check (checking the functional coverage), the selection process based on empiricism as well as the operation of the established application. The basis of the approach is a collection of semantically-annotated sources: the process model repository, the entity repository, a quality model repository and furthermore an experience factory.

The process model repository is the source for process models that serve as descriptions for the functionality of the aspired distributed system. Example for such processes can be e.g. didactical approaches descriptions [4].

An important source for empirical quality evaluations are quality models being provided by a quality model repository. The specification of a certain quality model is realized by selecting and weighting appropriate attributes.

The entity repository contains entities, their semantic description and their evaluation data regarding all defined quality attributes.

The selection and adoption of process models and quality models are difficult tasks which constitutes the need for guidance and support. Based on the work of Basili and Rombach the usage of an Experience Factory is proposed, that contains among others an Experience Base and Lessons Learned [5].

3 Runtime Adaptation

Runtime adaptation is performed at several points within the framework. In fact, that targets the experience-supported selection of an adequate process model, the experience-supported selection of an appropriate quality model as well as the functional entity selection.

3.1 Process Model Selection

The selection of an appropriate process model that defines the functional requirements for the parts of the later distributed system is the first step. Due to the fact, that such a choice can be a manual process, it should be supported by an experience factory providing knowledge and experiences - lesson learned - for the decision for or against a specific process model for the current need. The process model essentially base on semantic metadata to allow the later automatic mapping of semantically described entity functionalities to the functional requirements specified by the process model. According to [6] only formal descriptions of those models are applicable.

With the chosen process model a set of concrete distributed systems within the specified functional range is possible.

3.2 Quality Model Selection

The second step of the presented approach is a selection of a quality model from a quality model repository. This is intended to be done automatically. For certain domains manual adaptations can be more efficient. A manual individualization of this predefined set of quality attributes as well as of their importance weighting is also possible. For these purposes an experience factory can be helpful again.

As a result of this step a process model and importance-ranked quality attributes are defined. Thereby the quality-related aspects of the framework are adapted to the specific needs of the particular user.

3.3 Quality-Driven Entity Selection

With these process model and quality model information, process step three is able to determine whether enough available entities exist to provide an acceptable amount of functionality demanded by the process model. If there is no acceptable coverage after the negotiation subprocesses, then an abort probability based on already collected data can be computed. The user needs to decide whether he accepts the probability or not. If not the distributed system provision process will be aborted.

In the case of an acceptable coverage the runtime subprocesses can start. The first determines the next process step to be executed following the process model. Therefore information about the last process steps can be taken into consideration to optimize the next process step execution. Now, up-to-date entity information, their evaluation values as well as the data of the quality model are available to identify the best entity possible.

Following the defined necessities and given data the entity selection is formally described below. For the following formulas let PM be the chosen process model. Function $f^{funct}(PM)$ specified in Formula 1 is used to determine the set of entities E from the entity repository. Each of them can deliver the functionalities specified within the chosen process model (cp. Formula 2).

$$f^{funct} : \text{Process model} \mapsto \{\text{Entity}, \dots\}. \tag{1}$$

$$E = f^{funct}(PM). \tag{2}$$

Using the classic normalization approach presented in Formula 3 (normalizing to the interval from 0 to 1), the evaluation values $v_{i,j}$ of quality requirements j defined in the quality model must be normalised for each entity i. These $v_{i,j}$ are the measurement/simulation values to anticipate the optimal decision for the next process step.

$$v_{i,j}^{norm} = \frac{v_{i,j} - min(v)}{max(v) - min(v)}. \tag{3}$$

With the help of the weighted requirements matrix from the (maybe adjusted) quality model the last step - the identification of the optimal entity according to the empirical data and the quality model QM - can be performed (see Formulas 4 to 8). Formula 4 adjusts the normalized evaluation values to ensure proper calculation. If $v = 1$ describes the best quality level then no adjustments are necessary, otherwise a minimum extremum is desired and $1 - v$ must be calculated.

$$f^{mm}(v) = \begin{cases} v & \text{if a maximal } v \text{ is the best,} \\ 1 - v & \text{if a minimal } v \text{ is the best.} \end{cases} \tag{4}$$

$$f^{eval}(e_i) = \{\sum_{j=0}^{n-1} f^{mm}(v^{i,j}_{norm}) | e_i \in E \wedge n = |QM|\}. \quad (5)$$

$$V = \{f^{eval}(e_i) | \forall e_i \in E\}. \quad (6)$$

$$e^{worst} = e_{index}, index = min(\{x | v_x = min(V)\}) \wedge e_{index} \in E. \quad (7)$$

$$E' = E \backslash e^{worst}. \quad (8)$$

To determine the best evaluated entity, Formulas 5 to 8 are repeated until E' contains only 1 element. It provides the needed functionality and is the most appropriate one according to the specified quality model.

3.4 Process Model Types for Adaptation

The process models may vary in their basic structure according to the special, application-dependent requirements. According to this, their processing and thereby the automated adaption can be classified [7] Amongst others, the following types can be identified.

Sequential: Sequential process models are used for the modeling of sequential assembly and execution processes. Conditions are used to define functional decisions and to thereby create the adapted target system: maybe an adapted infrastructure, an e-Learning course or a measurement infrastructure.

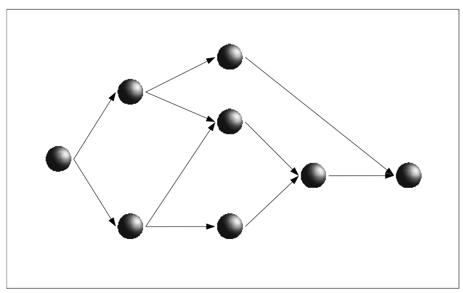

Figure 2. Sequential Process Models

Sequential with separated supervision: Sequential process models with separated supervision are used for the modeling of sequential assembly and execution processes, too. They additionally include downstream supervision process steps.

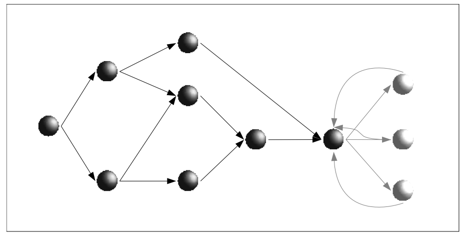

Figure 3. Sequential Process Models with Separated Supervision

Sequential with integrated supervision: Sequential process models with integrated supervision are similar to the one described above. In contrast, the supervision points back to the creation process steps.

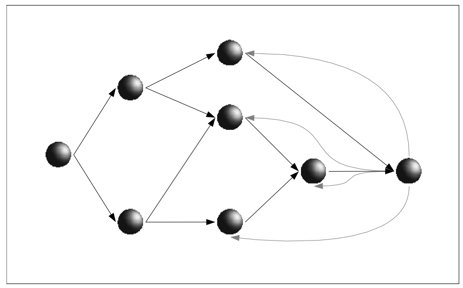

Figure 4. Sequential Process Models with Integrated Supervision

Supervision: Supervision process models only target the supervision of an existing system. Several conditions point away from a central event handling process step.

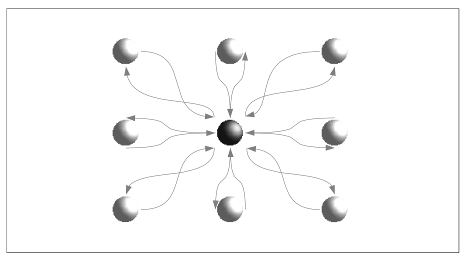

Figure 5. Supervision Process Models

Externally influenced: All types of process models being described above can be externally influenced by events outside the currently defined model. Thereby, meta-dependencies can be modeled.

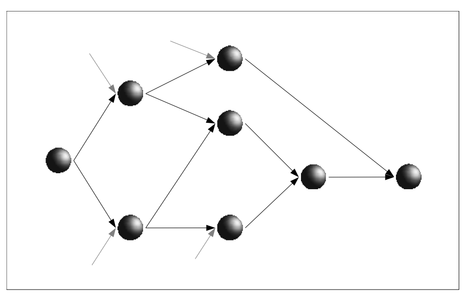

Figure 6. Externally Influenced Process Models

4 Conclusion and Further Work

The $QuaD^2$-Framework can be implemented using various technologies as e.g. ontologies, web services and agents. The presented quality-driven approach uses semantic descriptions for processes automation and supports different quality models and quality attribute evaluations.

Automatic quality measurement, evaluation and quality-driven entity selection within the general $QuaD^2$-Process are major building blocks for an high quality automatic runtime adaptation.

An implementation of this approach for specific systems is currently being performed. For the areas of e-Learning systems [2] and software measurement infrastructures [3] first components are realized.

References

1. Kunz, M., Mencke, S., Rud, D., Dumke, R.: Empirical-Based Design – Quality-Driven Assembly of Components. In: Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2008), Las Vegas, Nevada, USA (2008)
2. Mencke, S., Dumke, R.R.: A Hierarchy of Ontologies for Didactics-Enhanced E-learning. In Auer, M.E., ed.: Proceedings of the International Conference on Interactive Computer aided Learning (ICL2007), Villach, Austria (2007)
3. Kunz, M., Schmietendorf, A., Dumke, R., Wille, C.: Towards a Service-Oriented Measurement Infrastructure. In: Proceedings of the 3rd Software Measurement European Forum (SMEF 2006), Rome, Italy (2006) 197–207
4. Mencke, S., Dumke, R.: Didactical Ontologies. Emerging Technologies in e-Learning (iJET) **3**(1) (2008) 65–73
5. Basili, V.R., Caldiera, G., Rombach, H.D.: The Experience Factory. In Marciniak, J.J., ed.: Encyclopedia of SE. Volume 1. John Wiley & Sons (1994) 511–519
6. Mencke, S., Zbrog, F., Dumke, R.: Useful e-Learning Process Descriptions. In: Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST 2008). Volume 1., Funchal, Madeira, Portugal, INSTICC Press (2008) 460–463
7. Mencke, S.: Proactive Ontology-Based Content Provision in the Context of e-Learning. PhD thesis, Otto-von-Guericke University of Magdeburg (2008)