Leveraging Quality of Service and Cost in Cyber-Physical Systems Design

Christos Kotronis[1], Anargyros Tsadimas[1], Mara Nikolaidou[1], Dimosthenis Anagnostopoulos[1], George Dimitrakopoulos[1], Abbes Amira[2], and Faycal Bensaali[2]

[1] Department of Informatics & Telematics, Harokopio University, Athens, Greece
{kotronis, tsadimas, mara, dimosthe, gdimitra}@hua.gr
[2] College of Engineering, Qatar University, Doha, Qatar
{abbes.amira, f.bensaali}@qu.edu.qa

Abstract. Cyber-Physical Systems (CPSs) comprise multiple cyber-parts, physical processes, and human participants (end-users) that affect them, and vice versa. During the design of such systems, it is critical for the designer to take into account the end-user-perceived quality of provided services, as well as their cost, and integrate them into the CPSs; striking a satisfactory balance between quality and affordability is critical to system acceptance. In this work, we propose a model-based approach, using the Systems Modeling Language (SysML), to explore system design, encapsulating Quality of Service (QoS) and cost aspects, as system requirements, into a core model. Via this approach, the designer can define the system structure, configure it, measure and evaluate the quality, while analyzing cost, and find the best solution(s) for a correct design. As a use case, this approach is applied to a healthcare CPS, namely the Remote Elderly Monitoring System (REMS). In that context, managing REMS QoS and cost requirements, can contribute to an effective system design and implementation, enhancing the end-user satisfaction.

Keywords: Cyber-Physical Systems · Model-Based Design · SysML · Quality of Service · Cost Analysis · Remote Elderly Monitoring

1 Introduction

Cyber-Physical Systems (CPSs) are the integration of cyber parts, e.g., electronic system components like sensors, mechanical components, physical processes, as well as humans, i.e. the end-users that actively interact with them. During the design of such systems, the user-perceived quality of the provided services, as well as associated costs, should be taken into consideration, otherwise the system may perform poorly, inducing high expenses for its operation [2].

While several efforts have been made to design CPSs [8], the majority of them do not integrate and manage quality/cost from the early stages, i.e. from concept, to the final stages, i.e. system evaluation [10]. Among the available approaches, model-based design can facilitate the designer to define and evaluate high-quality systems, providing her the means to incorporate Quality of Service (QoS) [9] and cost in a core system model, configure it, and evaluate the system (based on QoS and cost). Doing this, she can consider trade-offs between quality, usability and affordability, and create an improved and satisfactory system.

In this work, we introduce a model-based approach to effectively design CPSs, evaluating them from a QoS and cost perspective. Specifically, using the Systems Modeling Language (SysML) [17], we integrate quality and cost aspects into a core CPS model, as system requirements. The designer can choose different system design configurations, verify these requirements, and consequently, evaluate the system; if the required objectives are not achieved, the approach enables the designer to explore alternative designs, reaching a better solution. The designer follows a novel iterative step-wise process, starting from the system definition, going to its configuration, and ending to its evaluation. During this process, quality and costs are defined, measured, and evaluated, enabling the designer to check whether the system can satisfy the end-user's needs.

The Remote Elderly Monitoring System (REMS) is employed as a case study for our CPS design approach; it is a healthcare system, used by elderly individuals to measure their vital signs, while their medical condition(s) is being monitored by professional health caregivers [1]. An effective design and implementation of such CPSs [7, 15] is crucial, since the elderly patients depend on high-quality healthcare monitoring services, while the system must remain affordable to them; focusing on the balance of quality and costs, results in greater satisfaction, and acceptance from the end-users. We stress that, in contrast to other works (see Section 2), our approach enables tuning this balance *by design*.

The paper is structured as follows. In Section 2, a short overview of related work is presented. Section 3 contains a description of our model-based approach, while in Section 4, this approach is applied to the REMS CPS use case. Finally, we conclude the paper and propose directions for future work.

2 Related Work

The need to design and implement CPSs in various domains has been expressed in [18]. In fact, CPSs should be designed –and implemented– with the goals of (i) providing adequate services to their end-users, while (ii) remaining affordable to them. Related efforts typically focus on these two goals separately [20]; however, their combination during all system stages is (a) crucial [3], and (b) remains an open challenge. For example, QoS is the quantitative index for the overall performance of provided services [9], thus it is important to be taken into consideration during the design, where it should be measured, evaluated, and preserved at a high level [16]. However, if the designer focuses solely on excellent QoS, ignoring the needed costs, the resulting system could be cost-prohibitive.

In [13], the estimation of the system's quality, requires suitable evaluation characteristics for design and analysis. An efficient way to manage quality, is via model-based approaches and a system model perspective [14], that can lead to efficient CPS design and development. While various model-based design techniques for CPSs [10] have been proposed, only few properly model and integrate critical requirements, like QoS, etc. [20] into the system. For example, the authors of [4] investigate specific QoS requirements in CPSs, while depicting some state-of-art CPS QoS models. In addition, in [6], adaptive CPSs are designed via quality requirements and parametric models, verified during system execution.

In summary, incorporating and evaluating *both* quality and cost still remains an unexplored area. To that end, our proposed model-based approach comes into play; we design CPSs, enriched with quality and cost aspects, verifying whether they are satisfied. Different solutions can be explored to reach a satisfactory system design that can balance the user-perceived QoS and cost affordability.

3 Integrating QoS & Cost Requirements into CPS Design

In this work, we focus on the effective design, configuration and evaluation of CPSs from a QoS and cost perspective, which are critical [20] in designing a system with maximal performance and user satisfaction. Specifically, our model-based approach facilitates the designer to construct a core system model, form its structure, and define QoS and cost requirements, that affect the system functionality. With those elements in place, she can explore alternative system design configurations, and evaluate the system, measuring and assessing QoS, while performing a cost analysis. External models can be integrated into the core model, providing additional information to tune or extend the defined model elements; this allows the designer to create a more practical and effective system model.

Having the CPS core model in the center, the designer performs specific actions, summarized in four stages; these form an iterative "design, configure, evaluate" process during system design. In each stage, the designer exploits constructs from the CPS model, while providing input elements –or additional models– to it. The stages are described, in detail, in the following.

Stage 1: Define the system's structure. The designer constructs the initial CPS model; she specifies the basic components that comprise the CPS, in an abstract fashion. Specifically, an abstract system components model, containing this structure, is created as a SysML Block Definition Diagram (BDD) [5].

Stage 2: Define (a) QoS requirements, (b) cost elements and related requirements. This is a crucial stage, since the designer integrates QoS and cost aspects into the system model. Regarding QoS, it is modeled as SysML requirements [5], within a SysML Requirement Diagram (RD) [5]. These requirements may obtain graded values, representing levels of the quality satisfaction; the desired level of each requirement is defined by the designer herself. In parallel, this stage provides an abstract costs model, comprising cost entities; system components estimate the respective cost entities, that are used to measure and hold the components expenses, e.g., their acquisition cost, etc. Following the creation of the costs model, the designer defines related graded cost requirements within a cost RD.

QoS and cost requirements are verified in following stages, based on their levels, allowing the designer to check whether the system provides high-quality services that can fully and efficiently serve its end-users, while remaining affordable to them. For this purpose, a verification model is applied to the CPS core model, providing verification elements that measure and evaluate QoS and cost.

During this stage, the designer can also create relationships between the requirements and any other model element. The system components satisfy the defined QoS requirements, via "satisfy" connections, while the cost entities satisfy respective cost requirements. In addition, the verification elements evaluate both system components and cost entities, and verify their corresponding requirements, via respective "evaluate" and "verify" connections.

Stage 3: Configure the system. In this stage, the designer exploits the components and cost models, and chooses a pre-defined configuration for the system, that is applied to them. The selected configuration is used to populate the components' properties with specific values that lead to the verification of the requirements, assisting the system evaluation. If the designer desires to, she can select the "best" configuration/solution, that satisfies all or the majority of the defined QoS and cost requirements; this configuration, populates the components' properties with values that correspond to the "best" system design.

Stage 4: Evaluate the system. At this –final– stage, the requirements are verified and the system is evaluated. To do this, the designer exploits the verification model, using the SysML Parametric Diagram (PD) [5], i.e. a construct within the verification model, in order to check the correctness and performance of the system design, as well as evaluate the system, assessing its QoS and cost. In case the latter are not satisfactory, the designer can explore alternative system designs, in order to reach a suitable solution. The PD's purpose is to exploit the configuration-generated properties, and calculate the real requirement level, that will be compared with the desired one. Requirements which are not verified during this process, failing to deliver the required level, are properly indicated in the modeling environment so that the designer can focus on improving them.

Approach summary. Figure 1 depicts the iterative design process, the CPS core model and the models/elements "exchange" between them. In each stage, the designer exploits constructs from the CPS model, e.g., abstract components model, and provides input, e.g., components model, enhanced with newly created properties and values, to it. Stages 1 and 2 may be considered as the "entities definition" stages, i.e. the designer defines the structure, QoS and cost entities and requirements, as well as verification elements. Stages 3 and 4 allow the designer to populate these entities with values, configuring and evaluating the CPS. The aforementioned input/output models/elements, along with their relationships, e.g., the components satisfying QoS requirements, create an integrated CPS design model, enriched with structural, QoS, and cost aspects. In summary, our approach enables the designer to: (i) define the system model, depicting the system's structure, (ii) focus on the system's QoS and cost requirements specification, as well as their verification, (iii) configure and evaluate the system model

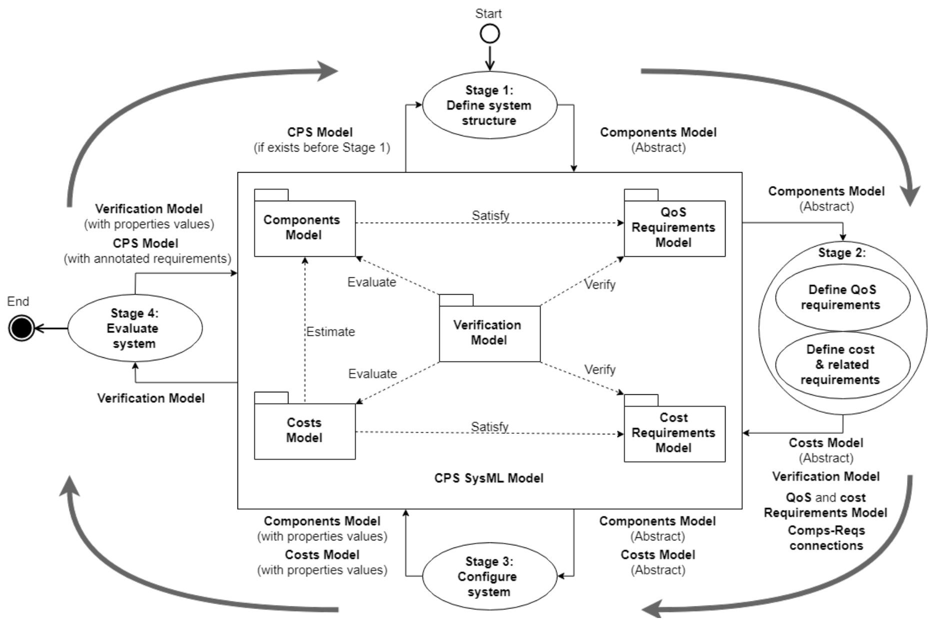

Fig. 1. Model-based approach for designing CPSs.

via formal methods such as parametric execution –via the PD–, and (iv) explore and decide on alternative design solutions/configurations.

4 Healthcare CPS: The REMS Case Study

In this section, we illustrate the feasibility of the proposed approach to the healthcare REMS CPS, following the associated designer steps. REMS is a representative implementation of a CPS that requires high-quality healthcare services and reduced costs, enhancing its performance and its end-users' satisfaction. As our case study, we focus only on the *Home* subsystem, where the elderly patient resides and operates the medical equipment in the context of the REMS CPS.

Define REMS Home subsystem structure. According to previous work [11, 12], basic CPS structural elements are the *Device*, representing mechanical or electronic components, the *Aggregator*, typically representing a central unit that collects the Device(s)-generated data, and the *Layer*, i.e. the middle level for the Device(s) and Aggregator connection.

Based on these, the designer creates the REMS Home model, specifying the following structural components: (i) an *Electrocardiogram (ECG)* Device, used to measure and monitor the elderly's heart rate, and detect heart attacks, arrhythmias, etc.; (ii) a *FallDetection* Device, useful for recording the patient's acceleration and orientation, monitoring her body position (e.g., falling or standing);

(iii) the Aggregator-type *IoTGateway*, that gathers and processes data, generated from the peripheral Device(s); (iv) the *ElderlyPatientHome* Layer, where the Device(s) and the Aggregator communicate. The latter is a composite component. Here, the ECG, the FallDetection and the IoTGateway are its respective parts, connected with *composition* relationships, forming the REMS Home's structure hierarchy. In Figure 2, an excerpt of the REMS Home subsystem model is depicted; the white-colored elements represent its components.

Define QoS requirements for REMS Home subsystem. Based on [12], specific QoS requirement types allow the designer to define graded QoS requirements in the REMS Home model. In particular, *Time*, *Security*, and *SWaP* [19] types can be exploited, along with properties that describe them, i.e. a unique *id*, a *text*, and the satisfaction *level* (described in Section 3).

In our case, Time requirements, regarding REMS's real-time behavior, e.g., real-time transmission of patient data from medical Device(s) to an Aggregator, are specified. Moreover, Security requirements, ensuring secured data, and SWaP, regarding the components' energy consumption, e.g., battery lifetime, and size properties, e.g, Device portability, are defined. Figure 2 illustrates the blue-colored *RealTimeTransmission* Time requirement, with id = "4", a text describing the need for this requirement, and the desired "real-time" level.

Define cost entities & requirements for REMS Home subsystem. To design REMS from a cost perspective, and integrate cost aspects into its model, cost entities and related requirements are specified by the designer. Considering a CPS's capital expenditures (CapEx) and operation expenses (OpEx), she can define *CapEx* and *OpEx* cost entities, along with a *value* property, to assess the system components' worth, and a *measurement unit*, i.e. the currency. Similarly to the QoS requirements, *Costing*-type requirements can be specified; cost entities must satisfy these requirements. For example, in Fig. 2, the white-colored *LayerCapEx* holds the Layer's value and currency; thus, the ElderlyPatientHome estimates this CapEx cost entity, which in turn satisfies the *LayerAcquisition* Costing requirement, regarding the components' overall purchase.

Define verification elements for REMS Home subsystem requirements. Along with the requirements' definition, the designer must designate elements, suitable to verify each requirement, and thus, assess desired QoS and cost levels. To do that, the real QoS (or cost) level value must be calculated, stored, and compared to the desired level. For this purpose, two verification elements are defined; one used for calculation, and the other for storing the real value. A *VerificationReqFormula* is created for each requirement, holding the expression –typically, an inequality– to calculate the real level. These expressions are primarily used in the requirements' verification process, thus, formulas refine the textual QoS/cost requirements. For example, Fig. 2 shows the formula for the patient data RealTimeTransmission level. If the corresponding property has "RT" value, then the output level is "real-time"; in case of delay, the value is "best effort" or worse.

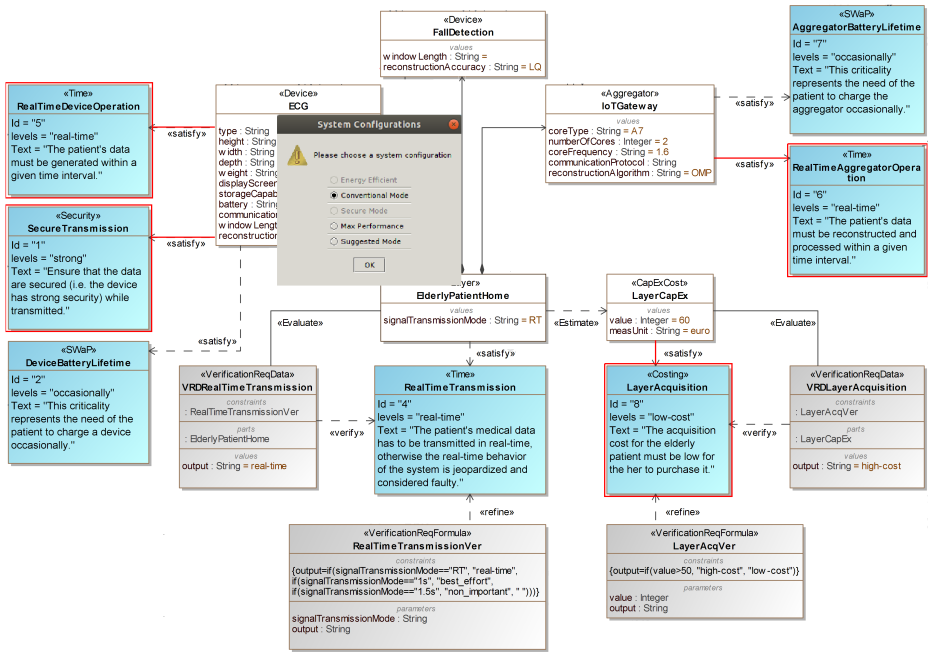

Fig. 2. REMS Home subsystem model configuration and evaluation.

Configure REMS Home subsystem. At this step, the designer configures the system, populating the components' and cost entities' properties with appropriate values; this leads to the system's evaluation. Specifically, the modeling environment provides a list of pre-defined configurations to the designer, allowing her to choose one for each component or the overall system. After her decision is applied, specific components' properties are automatically populated with pre-constructed values, regarding the chosen configuration. For example, in Fig. 2, the designer selects the "Conventional Mode" configuration; in this mode, among other features, patient data is generated, transmitted, and processed in real-time. Upon selection, values related to real-time behavior are automatically incorporated into its properties, i.e. Layer's *signalTransmissionMode* is "RT" (Real-Time), etc. These values are used as input to the VerificationFormulas expression, in order to calculate them and extract the real requirement level.

To assist the designer further, an external tool provides the "best" configuration option; this enables the automated calculation and pre-population of the model elements properties' values, so that all QoS and cost requirements are satisfied, and the designer-specified levels are achieved. When the designer selects this configuration, the calculated results return back to the modeling environment as values that effectively correspond to the "best" system design.

Evaluate REMS Home subsystem via its requirements verification. At this step, the designer exploits the VerificationFormulas, to calculate expressions, and the

VerificationReqData elements, as placeholders for the extracted real QoS/cost levels. The PD, mentioned in Section 3, is used to receive input parameters, insert them to a formula expression, execute it, extract the resulting value, and store it for further analysis (here, the system evaluation). Each PD is created within a VerificationReqData; this element's only property is the output of the expression's calculation. In addition, these elements verify corresponding requirements and evaluate the system components via respective relationships.

Finally, the –calculated– real requirement level is compared to the desired one, leading to the requirements' verification, and, thus, the evaluation of the system. In Fig. 2, the calculation of the formula that refines the RealTimeTransmission requirement, returned "real-time" as the output value, stored in the related VerificationData; this value is compared to the desired QoS level (also "real-time"). Since the real value is at least as good as the desired value, this requirement is verified. In parallel, regarding the cost requirement, the calculated "high cost" value is worse than the "low cost" desired level. Note that when a requirement is not verified, it is annotated in the model with a red-colored frame. In this case, the modeling environment alerts the designer, recommending appropriate actions she can take, like choosing another configuration.

5 Conclusions

In this work, we proposed a model-based approach to attack the challenge of integrating quality and costs into CPSs, during their design, as well as balance these concepts, in order to provide high-quality system services, while costs remain tolerable for the user. The approach comprises different steps that a designer can follow, enabling her to: (i) create a core system model, enriching it with structural elements, and QoS and cost aspects, in the form of requirements, (ii) configure the system, (iii) evaluate the system, via the verification of the requirements, assessing both quality and costs, (iv) exploit alternative design configurations, improving the user-perceived QoS, and consequently, improving the CPS. This approach was applied to the REMS, a healthcare CPS; the designer followed the workflow from the system concept and definition, to specific REMS QoS and cost requirements specification, to its configuration and evaluation. As future work, we plan to apply the approach to other CPS domains, where designing them from a quality perspective, is crucial for their users.

6 Acknowledgments

The authors wish to acknowledge Qatar National Research Fund project EMBIoT (Proj. No. NPRP 9-114-2-055) project, under the auspices of which the work presented in this paper has been carried out.

References

1. Baig, M.M., Gholamhosseini, H.: Smart health monitoring systems: an overview of design and modeling. Journal of medical systems **37**(2), 98 (2013)
2. Brauner, P., Valdez, A.C., et al.: On studying human factors in complex cyber-physical systems. Mensch und Computer 2016–Workshopband (2016)
3. Broy, M., Schmidt, A.: Challenges in engineering cyber-physical systems. Computer **47**(2), 70–72 (2014)
4. Dillon, T., Potdar, V., Singh, J., Talevski, A.: Cyber-physical systems: Providing quality of service (qos) in a heterogeneous systems-of-systems environment. In: 5th Digital Ecosystems and Technologies Conference. pp. 330–335. IEEE (2011)
5. Friedenthal, S., Moore, A., Steiner, R.: A practical guide to SysML: the systems modeling language. Morgan Kaufmann (2014)
6. García-Valls, M., et al.: Pragmatic cyber physical systems design based on parametric models. Journal of Systems and Software **144**, 559–572 (2018)
7. Haque, S.A., Aziz, S.M., Rahman, M.: Review of cyber-physical system in healthcare. international journal of distributed sensor networks **10**(4), 217–415 (2014)
8. Hehenberger, P., Vogel-Heuser, B., et al.: Design, modelling, simulation and integration of cyber physical systems: Methods and applications. Computers in Industry **82**, 273–289 (2016)
9. ISO: Iec 25000 software and system engineering–software product quality requirements and evaluation (square)–guide to square (2005)
10. Jensen, J.C., Chang, D.H., Lee, E.A.: A model-based design methodology for cyber-physical systems. In: 2011 7th International Wireless Communications and Mobile Computing Conference. pp. 1666–1671. IEEE (2011)
11. Kotronis, C., Nikolaidou, M., et al.: A model-based approach for managing criticality requirements in e-health iot systems. In: IEEE 13th Conference on System of Systems Engineering (SoSE). IEEE (2018)
12. Kotronis, C., Routis, I., et al.: A model-based approach for the design of cyber-physical human systems emphasizing human concerns. IEEE International Congress on Internet of Things (2019), to be presented
13. Laso, P.M., Brosset, D., Puentes, J.: Monitoring approach of cyber-physical systems by quality measures. In: International Conference on Sensor Systems and Software. pp. 105–117. Springer (2016)
14. Liu, Y., Peng, Y., et al.: Review on cyber-physical systems. IEEE/CAA Journal of Automatica Sinica **4**(1), 27–40 (2017)
15. Monisha, K., Babu, M.R.: A novel framework for healthcare monitoring system through cyber-physical system. In: Internet of Things and Personalized Healthcare Systems, pp. 21–36. Springer (2019)
16. Mourtzis, D., Vlachou, E.: Cloud-based cyber-physical systems and quality of services. The TQM Journal **28**(5), 704–733 (2016)
17. Object Management Group - SysML: System Modeling Language (2018), `https://www.omg.org/spec/SysML/`
18. Sanislav, T., Miclea, L.: Cyber-physical systems-concept, challenges and research areas. Journal of Control Engineering and Applied Informatics **14**(2), 28–33 (2012)
19. Thales Defense & Security, I.: Design considerations for size, weight, and power constrained radios. 2006 Software Defined Radio Technical Conference and Product Exposition (2006)
20. Zheng, C., Le Duigou, J., et al.: Multidisciplinary integration during conceptual design process: A survey on design methods of cyber-physical systems. In: DS 84: 14th International Design Conference. pp. 1625–1634 (2016)