

Evaluation of Dynamic Instantiation in CPRM-Based Systems

Ana Nieto^(✉)

Computer Science Department, University of Málaga, Málaga, Spain
nieto@lcc.uma.es

Abstract. Context-based Parametric Relationship Models (CPRMs) reduce the complexity of working with various numbers of parameters and dependencies, by adding particular contexts to the final scheme when it is required, dynamically. In this paper the cost of including new information in CPRM is properly analysed, considering the information in the parametric trees defined for the parameters in the CPRM-based system. Some strategies for mitigating the cost of the instantiation process are proposed.

Keywords: CPRM · Security · QoS

1 Introduction

Security and Quality of Service (QoS) mechanisms are fundamental to providing a total network convergence. The convergence of networks poses several challenges regarding QoS and security. Most of these are inherited from the traditional tradeoffs in isolated networks, but the *Internet of Things* (IoT) and the *Future Internet* (FI) open the door to new challenges to be met.

For example, the user's participation in these networks, requires providing multimedia capabilities that, in many cases, may be provided through limited-resource devices, such as sensors acting as relays. Moreover, the user's participation requires taking into account misbehaviour issues, and security mechanisms should be applied. The problem is that security mechanisms require network and local resources that the QoS mechanisms handle to guarantee the performance of services. Moreover, in future networks, multipurpose devices may have to coexist. That means that, on the one hand, different capabilities can be provided regarding the local resources and functionality of the devices. On the other hand, some devices in the network have different purposes and therefore are not available for providing security or QoS requirements.

Consequently, in the IoT and FI, security and QoS tradeoffs cannot be considered, only taking into account specific scenarios, but also have to consider abstract issues or generic composition of *things*. Diverse models for measuring the security and QoS tradeoff have emerged [1–3]. In particular, from our point of view, the *generic models for the analysis of Security and QoS tradeoff* are well suited to be used in heterogeneous networks of dynamic composition, in where

it is very difficult to predict, with any great accuracy, the devices that will form the network. An example of these types of models are *Context-based Parametric Relationship Models* (CPRM).

The main objective of this paper is to evaluate, the impact of the instantiation of parameters in CPRM-based systems. That is, provides a discussion about the complexity of the models based on the number of dependencies, and the location of the instantiated parameters in the general tree, and draw conclusions about how the effect on performance may be mitigated in order to enhance the implementation of the instantiation process carried out in CPRM-based systems.

The paper is structured as follows. Section 2 presents the state of the art in generic models for the analysis of the security and QoS tradeoff. Section 3 provides an overview of the CPRM model. Section 4 defines the impact of the instantiation in CPRM-based systems, according with the definitions provided by the model. Finally, Sect. 5 proposes some new directions that could be taken to mitigate the impact on performance.

2 Background

Generic models for assessing the Security and QoS tradeoff are those capable of analyzing the security and QoS requirements and characteristics of a set of elements and components in a network, being able to change the composition of said elements and characteristics for other ones and still be useful. Therefore, the idea behind these models is that a part of the model has to remain abstract prior to knowing or receiving the new components in the information system.

Related work that focus on security and QoS tradeoff consider the composition of services based, for example, on the selection of a set of security goals that may be analysed based on their interdependencies [2], or the definition of ontologies where security is taken as a static metric for the QoS [4], whereas, our perception, is that security can be very dynamic, based on the context, and, moreover, it is not always possible to predict the final mechanisms that will be available in the network.

Moreover, there are additional approaches that have provided interesting results but cannot be easily integrated in static approaches. For example, security and QoS are analysed as separate issues in [5,6]. In [5] an analysis of QoS and *Quality of Experience* (QoE) is provided, while in [6] authentication protocols for mobile devices are analysed. Despite the fact that both of them focus on next generation networks, the possibility of comparing them and drawing conclusions is very difficult without using generic models for the analysis. Furthermore, security and QoS analysed as separate issues is useful for providing rich information to be added to the former, and, then, use rule-based parametrization techniques for defining different compositions, at service layer [7]. However, these approaches focus on services, and do not consider the composition of *things* (e.g. combination of anti-tampering mechanisms and high-layer services).

Generic approaches for analyzing complex decisions such as *Analytic Hierarchy Process* (AHP) [8], or *Potentially All Pairwise Rankings of possible Alternatives*

(PAPRIKA) [9] are fixed and do not define contextual parameters as CPRM does. Moreover, the purpose of CPRM is the analysis of security and QoS tradeoff based on the composition of mechanisms, and for this reason CPRM defines the set of relationships required to perform this type of analysis and to vary the subjective value of the parameters based on the context.

3 Contex-Based Parametric Relationship Models

The structure of a CPRM-based system, defined in [10], is based on a set of parameters and the relationships between them, a set of operations (*op*) which define the effects on the dependent parameters, and a set of weights which define the relevant subjective and non-subjective components in the model. To do that, the model defines three types of structures, depending on the type of context integrated:

- *Parametric Relationship Model* (PRM) structure. If only general parameters and their relationships are defined. The default relationships are defined for Security and QoS parameters.
- *Context-based PRM* (CPRM) structure. If there is a general context integrated.
- *Instance of CPRM/CPRM_i* (*CPRM_i*) structure. If there are one or more particular contexts integrated. The definition of this structure is recursive, and defines the dynamic behaviour of CPRM-based systems.

Therefore, a CPRM-based system is defined based on the parameters and relationships, using one structure of type CPRM or *CPRM_i*. The particularity of these systems is that we can change the context dynamically to evaluate the new mechanisms which operate in these systems.

With this objective in mind, there are two structures (scripts) for defining the contexts:

- *General Contexts* (GC), which define the weights for the elements in a PRM or a CPRM (parameters, types, layers, operations, relationships).
- *Particular Contexts*(PC), which define new parameters (instances) that instantiate to existing parameters in the model structure (*CPRM/CPRM_i*). The existing parameters are targeted with the type *instantiated* when the instantiation process concludes.

Contexts can be integrated into schemes to define the behaviour of the CPRM-based systems. Figure 1 illustrates the dynamic generation of CPRM-based systems according the description provided in [10], that should be implemented by any handler of CPRM-based systems. First, using a PRM the generic relationships between parameters are provided. In our case, the model is defined to be used for assessing the security and QoS tradeoff, so the set of parameters that is the result of diverse security and QoS analyses is used. The set of parameters chosen are general properties (e.g. authentication) or parameters that can

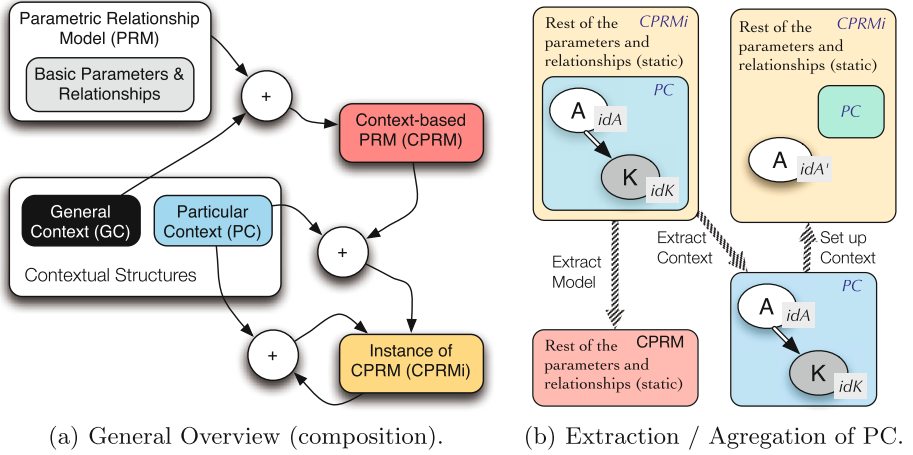


Fig. 1. Dynamic Generation of CPRM-based Systems.

be expressed using mathematical formulation (e.g. delay). So, using these parameters, it is very easy to understand the concept of *instantiation*: the goal is to replace the parameters with the specific mechanisms which implement them, when this information becomes available.

In consequence, the composition of a CPRM-based system is dynamic, and the objective is the evaluation of the mechanisms for implementing new properties as soon as they are known. Note that this is an approach for measuring the security and QoS tradeoff that respects the uncertainty in the composition of dynamic networks, where it is very difficult to predict the mechanisms provided by the heterogeneous devices which compose the network.

Moreover, the model defines a set of action rules for adding dependencies between the parameters once the instantiation process has been initiated. Figure 2 shows an example of instantiation of parameters. For example, the parameters A, B, and C were defined in the initial PRM, as well as the dependency $d(A, B)$. However, the parameters a1, b1, and c1 are instances, defined in the PC, and instantiate to the parameters A, B and C, respectively.

As a PRM is considered to be the most general structure, and, therefore, provides all general information, the relationship $d(a1, c1)$, defined in the PC, generates an inconsistency in the model, because information $d(A, C)$ was not previously defined. So, the action rules are defined to consider these cases and add these kinds of relationships between instantiated parameters.

Furthermore, any instance inherits the relationships of the instantiated parameter. Then, as $d(A, B)$ has been previously defined in the general behaviour, $d(a1, b1)$ inherits this behaviour with the same weight as defined in the general relationship. In fact, the relationship $d(A, C)$ was added with weight 0 to avoid the effect of this last action rule.

In addition, during the instantiation process, the system checks the identifier of the parameters to prevent different parameters from being considered as the

Weights: parameters (Wp), operations (Wo), types (Wt), layers (Wl), dependence between parameters (Wd)

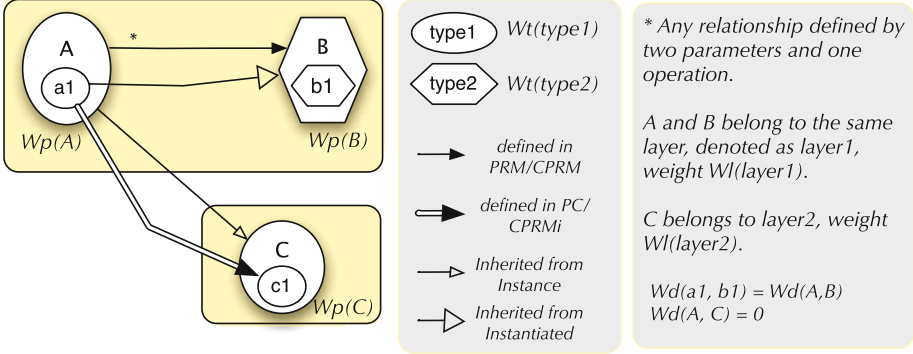


Fig. 2. Weight-based Relationship and instantiation.

same parameter, or to identify when a parameter in a PC extracted from a model has a different identifier in another contextual model where the PC has to be integrated. This latter case is shown in Fig. 1(b).

After the integration process, the model defines a set of operations that can be performed on the information to analyse the security and QoS tradeoff. Specifically, the set of operations are defined in Table 1. These operations are responsible for the final parametric tree for a parameter. For example, if the parameter A defines the relationship $A \xrightarrow{\pm} B$, and A is increased, then, the effect on B according to the dependence and Exp. 1 is an increase in B. To the contrary, if A is decreased, then, there is nothing to do, if the only information available is that $A \xrightarrow{\pm} B$, because in this case, only the effect when A is increased, is defined.

Therefore, a parametric tree is a tree that shows all the parameters affected by an increase or (exclusive) decrease of a parameter. The initial dependencies are set as a graph, so the cycles are avoided by creating branches when loops are found. Moreover, the rest of the leaves of the model are generated when the operation defined for the relationship does not show an effect on the parameter in the consequent.

Finally, consider that the cost of this dynamic behaviour is memory and computational. This is inevitable when considering the integration of new information in a cross-layered dependencies model. Measuring the cost based on the new

Table 1. Operations in a CPRM.

Basic Formulation Set (BFS)	Complex Formulation Set (CFS)
$D^+ :: aD^+b \Rightarrow (\Delta a \rightarrow \Delta b)$	(1) $D^c :: (\Delta a \rightarrow \Delta b) \wedge (\nabla a \rightarrow \nabla b) \equiv aD^+b \wedge aD^{-+}b$ (5)
$D^- :: aD^-b \Rightarrow (\Delta a \rightarrow \nabla b)$	(2) $D^t :: aD^c b \wedge bD^c a$ (6)
$D^{-+} :: aD^{-+}b \Rightarrow (\nabla a \rightarrow \nabla b)$	(3) $D^{-c} :: (\Delta a \rightarrow \nabla b) \wedge (\nabla a \rightarrow \Delta b) \equiv aD^-b \wedge aD^{-+}b$ (7)
$D^{-+} :: aD^{-+}b \Rightarrow (\nabla a \rightarrow \Delta b)$	(4) $D^{i+} :: (\Delta a \rightarrow \Delta b) \wedge (\nabla a \rightarrow \Delta b) \equiv aD^+b \wedge aD^{-+}b$ (8)
	$D^{i-} :: (\Delta a \rightarrow \nabla b) \wedge (\nabla a \rightarrow \nabla b) \equiv aD^-b \wedge aD^{-+}b$ (9)

information provided is vital in order to evaluate the suitability of the model for different purposes.

4 Impact of a Particular Context

The aim in this section is raise awareness of the impact of a PC based on the location of the instantiated parameters in the dependencies graph. When the PC defines a type of sensor or device, it is usual that the parameters involved will be leaves or parameters with low accumulative dependence. In this case, the parameters in the consequent are instantiated parameters whose impact on the rest of parameters is null. Hence, the impact of the relationships defined for these parameters, when they are set up in our model do not provide much information as regards from where there are extracted. So, the impact of a new PC is greater, as the new parameters defined (instances) affect parameters far away from the leaf nodes, and with a high number of relationships.

Table 2. Set-based definitions in a CPRM-based system [10].

Acumulative Influence (ι) and Acumulative Dependence (δ)	
$\iota(a) = I_a , I_a = \{x x \rightarrow a \vee xRa, x \neq a, x \in P\}$	(10)
$\delta(a) = D_a , D_a = \{y a \rightarrow y \vee aRy, y \neq a, y \in P\}$	(11)
$xRy \iff x \rightarrow y \vee \exists k k \in D_x \wedge k \in I_y$	(12)

This conclusion can be analysed using the formulation in Table 2. Considering N parameters in a CPRM, and Y the parameter that will be instantiated by K number of instances, which means that there are k parameters which satisfy that Y is their parent: $K = |\{x|Y \in P(x)\}|$. This set is known as the set of instances of a parameter Y , and is denoted as H in Exp. 14. Moreover, P defines the set of parents of a parameter, which is the information provided by the PC. Whether Y is a leaf node, that is, $\delta(Y) = 0$, and the accumulative influence on Y preceding the instantiation of the CPRM is $\iota(Y) = M$, then the new accumulative influence on Y after the instantiation is $\iota(Y) = K * M$, considering that Y is the only instantiated parameter.

Moreover, if the instantiated parameter Y is not a leaf node, it impacts on other parameters in the PRM ($\delta(Y) > 0$). So, in case that prior the instantiation the accumulative dependence on Y was $\delta(Y)$, and that $\delta(Y)^t$ defines the dependence degree of Y once the instances 1 to $t \leq K$ have been added, and therefore $\delta(Y)^0 = \delta(Y)$, after the instantiation the accumulative dependence on Y is given by Exp. 13 (δ'):

$$\delta(Y)' = \delta(Y) + \sum_{i=1}^K (\delta(y_i) - \delta(Y)^{i-1}) \quad (13)$$

$$H = \{x|Y \in P(x)\}, K = |H| \quad (14)$$

Therefore, the new accumulative dependence is calculated based on the new dependencies that are included because the instances can define new relationships, so new parameters can be affected. Therefore, the complexity when a new PC is added depends on the number of parameters and relationships but also the location of the instantiated parameters in the general parametric tree.

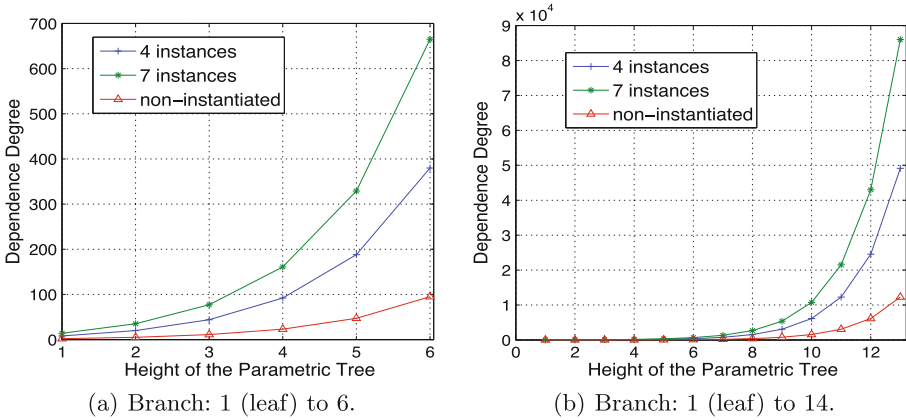


Fig. 3. Instantiation of one Parameter

Figure 3 shows the increasing number of dependencies (dependence degree) for a parameter regarding the length of the branch, considering 6 as the length of the longest branch in Fig. 3(a) and 13 in 3(b). The problem is simplified considering that the new instances only inherit the relationships defined by their parents, and do not define new relationships, and that the number of dependencies per parameter is fixed at 2.

Considering these restrictions, note that, although the instances do not define new dependencies, the number of dependencies for the parameters at upper layers increases. According to Table 2, the accumulative dependence is higher in those parameters far away from the leaves. Specifically, it depends on the position of the parameter and the number of dependencies behind it. When the parameter is instantiated, the accumulative dependence increases if new parameters appear in the parametric tree as a result of the new information provided in the instantiation.

Furthermore, the values chosen to show the previous results have been chosen only for testing purposes. Indeed, in a PRM, the number of dependencies, parameters and instances is free. For example, Fig. 4, shows the length of the parametric trees in a PRM with real parameters used for testing, where the longest branch may vary considerably, depending on the parameter, precisely because there is no fixed limit for the number of relationships. Moreover, in Fig. 4, two types of results are shown: the results for the increasing and decreasing parameters may generate different parametric trees according to the formulation in Table 1 and the

weights. If a dependence is weighted 0, then the effect of the parameter that is in the antecedent of the dependence is not propagated by the tree.

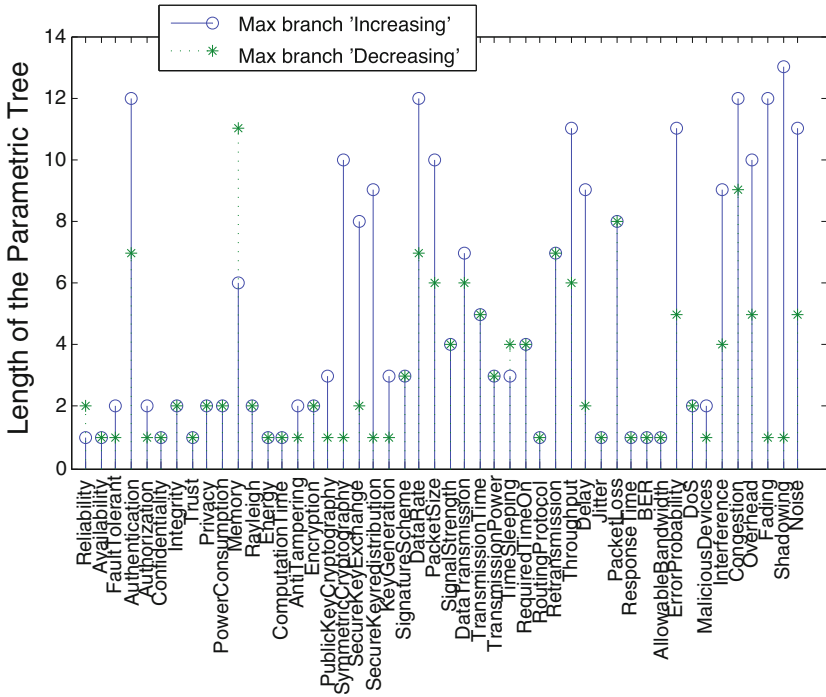


Fig. 4. Example: Length of Parametric Trees in a CPRM-based system.

Finally, the accumulative dependence degree depends on the number of dependencies defined for the system, and the instantiation may add new dependencies not defined by the parents. It must be remembered that Fig. 3 has been built, taking into account that the instances do not define new relationships. When new relationships are defined in the PC, which is generally the case, the impact on the final accumulative dependence increases. Moreover, it must be appreciated that, even if the instances only inherit the behaviour of their parents and do not include new dependencies, the calculation of the impact on the model of the instantiated parameters takes more time, because the process is repeated per instance.

For example, when the accumulative dependence increases, the complexity of the integration of parameters carried out by a handler of CPRM-based systems, is higher. An intuitive conclusion is that a possible way to reduce the time of integration of PCs, is to select the PCs which describe the physical layer components from the very beginning, because, in general, the physical description of components has a low dependency degree, when it concentrates on, for example,

the characteristics of the battery. These components are usually instantiated with instances that are affected by other components (appear as consequents in the formulation).

However, given the nature of our model, this is not always the best choice, because new relationships, that is, new behaviour, defined in PCs may completely change the behaviour of the final $CPRM_i$.

In other words, if the instances with new relationships, not defined in the model for the instantiated parameter, are not integrated at the beginning, and the instantiated parameter has a large number of instances, then, the new relationships have to be added to all the instances. The cost for this is very high, because it implies triggering action rules to maintain the coherence in the model, so, the whole parametric system is checked again when some incoherence is found. If the instances with new relationships are added at the beginning of the instantiation process, then, this new behaviour is taken by the parents which transfer the sum of the whole behaviour learned (the parent's behaviour and the new relationships) to the instances. Note that this criteria concerns the order in which the instances in the PC are to be set up in the model, so it can be done independently from the order in the selection of PCs or parameters to be instantiated in the CPRM.

As a result, the instantiation process may be enhanced to mitigate the adverse effects which impact on performance, considering:

- R1. Dependency degree and accumulative dependence of parameters to be instantiated.
- R2. The effect of the changes on the rest of the system (length of the max. parametric tree).
- R3. The order of the new instances and relationships defined in PCs, because the new behaviour may totally change the final $CPRM_i$.

5 Classifications and Mitigations

In this section, the focus is to provide a classification based on the information in the CPRM-based system and the type of the PC that will be integrated in order to adapt the integration process, to mitigate the effect on the performance because of the dynamic instantiation. Some conclusions that are drawn from this analysis are summarized in Table 3, where order means the priority in the instantiation process.

In order to test these conclusions, the CPRM-based system described in Table 4 is processed taking into account Table 3, to mitigate the effect in the integration of the PC, that is also structured according to the recommendations in Table 3¹. Note that the only relationships that are marked with a weight are those defined by the instances to make changes in the behaviour of the model.

¹ Information about default weights omitted for sake of clearness. It is assumed that the relationships that are completely new have weight 1.

Table 3. Recommendations in the Integration process.

Component	Characteristic	Order
CPRM	Parameters with shortest branch first (SBF)	1
CPRM	Parameters with max numer of instances	2
PC	Parameters that define new behaviour: new relationships to be added that maximizes $\max \delta'$	1
PC	Parameters that maximizes the final longest branch (if available)	2

Table 4. CPRM and PC definitions.

Parameter	Direct relationships	Branch	Instances	New relationships
A	3: B,C,E	6	A1, A2	-
B	0	1	B1, B2, B3	$B3 \rightarrow F1$
C	0	1	C1, C2, C3, C4	$C4 \rightarrow F1$
D	3: B,C,E	8	D1	-
E	2: F,I	7	E1, E2	$E1 \rightarrow C3, E1 \rightarrow B2$
F	1: H	3	F1	-
G	1: H	2	G1, G2, G3	$G1 \xrightarrow{w=0}, H1, G1 \xrightarrow{w=3}, H2$
H	0 -	1	H1, H2	-
I	2: J,K	9	I1	$I1 \xrightarrow{w=2} J1, I1 \xrightarrow{w=0} K2$
J	1: L	8	J1,J2	-
K	0 1	1	K1,K2,K3,K4	-
L	1: M	7	L1,L2	$L2 \xrightarrow{w=4}, M2$
M	2: J,A	6	M1,M2	$M1 \xrightarrow{w=2} J1$
N	1: B	2	N1	$N1 \rightarrow A, N1 \xrightarrow{w=2} B3$

For example, I is related to J and K according to Table 4. Then, the instances of I inherit the relationships of I , which means that they will be related to J and K , and, therefore, with their instances. However, the relationship $I1 \xrightarrow{w=0} K2$ redefines the weight in the relationship inherited to avoid the relationship between the instances $I1$ and $K2$. These relationships do not add additional relationships to the model, because the parents of the instances are already related with each other, so it is unnecessary to apply action rules. Therefore, the relationships that add new information in the model are $B3 \rightarrow F1$, $C4 \rightarrow F1$, $E1 \rightarrow C3$, $E1 \rightarrow B2$, and $N1 \rightarrow A$. These requires, respectively, to include the relationships $B \rightarrow F$, $C \rightarrow F$, $E \rightarrow C$, B and $N \rightarrow A$ to be included.

Consider that the order $A - N$ corresponds to the unsorted or default distribution where letters describe non-instantiated parameters and the instances for each parameter are described using the letter of the parent and a number. Note that one of the recommendations in Table 3 suggest the *Shortest Branch*

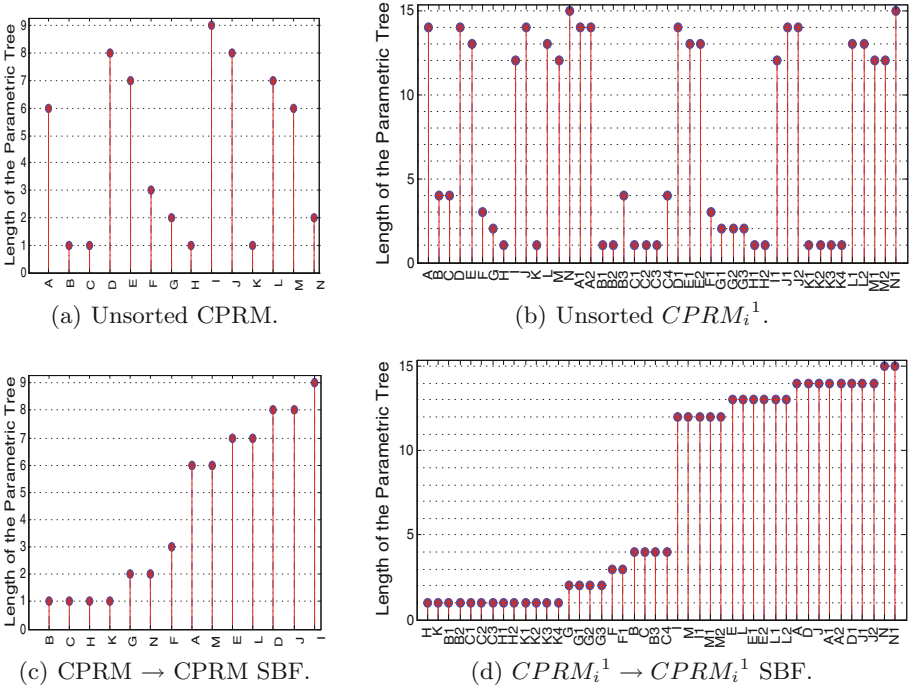


Fig. 5. Changes in the Parametric Trees after the Instantiation

First (SBF) criterion, that is the opposite to the *Longest Branch First* (LBF) criterion. Using the CPRM and the PC, 9 $CPRM_i$ are generated:

- $CPRM_i^1$: Unsorted CPRM. Unsorted PC.
- $CPRM_i^2$: Unsorted CPRM, PC SBF.
- $CPRM_i^3$: Unsorted CPRM, PC LBF.
- $CPRM_i^4$: CPRM LBF, Unsorted PC.
- $CPRM_i^5$: CPRM LBF, PC SBF.
- $CPRM_i^6$: CPRM LBF, PC LBF.
- $CPRM_i^7$: CPRM SBF, Unsorted PC.
- $CPRM_i^8$: CPRM SBF, PC SBF.
- $CPRM_i^9$: CPRM SBF, PC LBF.

Figure 5 shows the length of the branches of the original CPRM (before the instantiation process) and $CPRM_i^1$, before and after the instantiation. Note that a simple ordering in the CPRM before the instantiation is not sufficient, because the PC adds changes in the behaviour of the model that have to be considered. Indeed, the order of the parents (parameters of type *instantiated*) before and after the instantiation is not the same. For example, in Fig. 5(c), in a CPRM SBF it is recommended *C* before *G*. However, as can be seen in

Fig. 5(d), the instance $C4$ introduces more changes than the instances of G . So, this information cannot be considered if only the parameters in the CPRM are taken into account. Specifically, Fig. 5(d) has been built from Fig. 5(b), but this order is different from the final order that can be generated following the recommendations in Table 3, they may vary depending on the recommendations.

In order to check the different alternatives and the suitability of the recommendations, the instantiation process (add CPRM to PC to generate $CPRM_i$) is repeated 60 times per $CPRM_i$, and the average execution times for these processes are shown in Fig. 6². Note that the CPRM and PC chosen as seeds, integrate diverse relationships, defined only in order to force the maximum number of operations to action rules.

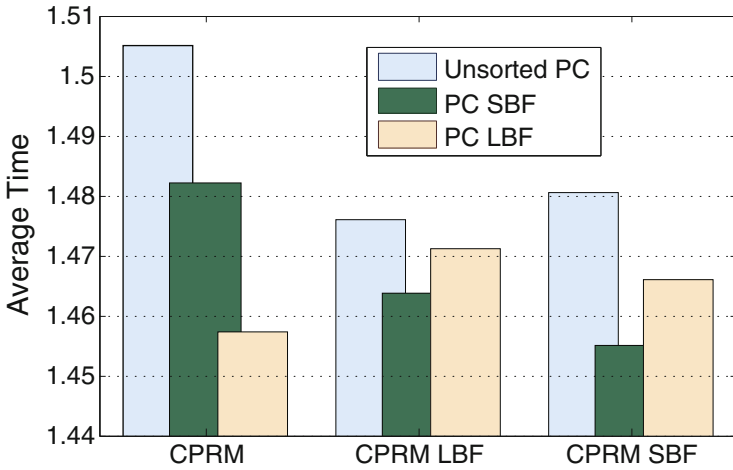


Fig. 6. Average Times in $CPRM_i^{1-9}$ Calculation.

Considering the results, when restrictions in Table 3 are applied to the CPRM before the instantiation, the performance (measured in terms of computation time) is improved with respect to the case where the CPRM is not sorted, or is sorted according to LBF. Moreover, in this case, the PC does not integrate a high number of relationships. Despite this, given the results, it is possible to appreciate that the advantages for sorting the parameters in the PC are conditioned by the max branch of the parameter that will be instantiated and the number of new relationships that the parameter introduces in the CPRM, and, therefore, the length of the new branches that have been added as a consequence of the

² Results calculated using MATLAB and our handler of CPRM systems for Security and QoS Tradeoffs (SQT). During the process, the handler plots different graphs and information for testing. Note that this additional functionality increases the overall time in all the integrations.

instantiation process. Moreover, as Figs. 5(c) and (d) show, the order in CPRM LBF and PC LBF or CPRM SBF and PC SBF is not the same, because the order in the PC is generated considering the new information that will be integrated in the model. It is the reason because CPRM LBF + PC LBF is not optimal, instead CPRM LBF + PC SBF. Note that, in the absence of informacin (case CPRM), the integration of the PC LBF is the best option, in this case.

It can be observed that the effect when sorting the parameters (instances) in the PC is higher than the effect in sorting the parameters in the CPRM. This is precisely because the integration process is costly, and, especially so, when the instances and relationships define a behaviour that is very different from the behaviour defined by the parents. It is expected that new instances inherit most of the behaviour of their parents, so in these cases maybe a better definition of the properties of the system to be instantiated, or a new classification in different layers of the parameters would be very helpful in mitigating the effect of the instantiation process. The knowledge of the system can be very helpful in order to identify the order of the parameters in a PC, because, when the environment is very dynamic, it is not always possible to define the adequate order in the integration of PCs or the order of their componets.

Finally, CPRM schemes are used to identify security and QoS tradeoffs. In previous work, a set of parameters and relationships which define general behaviour of security and QoS parameters extracted from a wide range of studies and mathematical formulation was provided [10]. With this purpose, the original scheme used to compose these systems, classifies the parameters based on their type and abstract layers. The classification of types includes: security (e.g. authentication), performance (e.g. delay), QoS (e.g. streamming), resources (e.g. battery), characteristics (e.g. type of antenna), *Quality of Experience* (e.g. mechanisms for measuring the user's experience), etc.

The objective then, is to have different PCs and instantiate the CPRM when the information about the final composition of the system becomes available. Given the previous results, and our classification, the integration of PCs should be performed as follows, based on the dynamisms of the parameters and our focus³:

1. High-layer relationships. Security and QoS requirements are described at this layer. So, given the nature of our analysis, the parameters in this layer should be instantiated at the end of the process, because, probably, they change frequently.
2. Local properties. At this layer there are described the resources in the devices of the network and the characteristics. For example, this layer describes physical characteristics of the components. So, at this layer there are several parameters that are leafs in many parametric trees.
3. Communication and Measurements layers. The most of the parameters defined at these layers are of type performance. These parameters are the most of

³ These are conceptual/abstract layers, not physical layers.

them defined based on mathematical formulation, and are influenced by other parameters at the same layer or from other layers. It is expected that the instances for the parameters at this layer do not include additional relationships that affect to the behaviour of the model. It is expected that the new changes redefine the weight to some parameters and relationships.

4. Environmental conditions. The effect of the environment in the system is described at this layer. As these are restrictions given by the environment, some of these parameters may not vary during a long time (e.g. the probability for wireless eavesdropping in a wired system; average of devices in an office, etc.). These parameters depend on the dynamism of the system.

Moreover, the preferences in the selection/integration of PCs, should be adapted based on the whole information about the system where this mechanism will be used. In general, this solution may be useful when the system provides a rich variety of information that is stored in data bases and may be defined using parametric relationships. Then, the integration/extraction of contexts may help to understand the different problems in the integration of security and QoS mechanisms in the environment, before their deployment.

6 Conclusions and Future Work

The Context-based Parametric Relationship Model (CPRM) enables the composition of parameters that may be instantiated based on the context. The instantiation consists in providing mechanisms for general parameters, which are defined in the model. It is a fact that when the number of parameters increases, so does the complexity of the CPRM-based system. In these models, the complexity can be analysed, using the size of the parametric tree defined for the parameters to be instantiated. In this paper, the impact on the influence and the dependence degree based on the position of the instantiated parameter in the dependencies tree has been discussed, and alternatives to mitigate this impact are proposed and analysed.

In addition, a problem beyond the scope of this paper is that the dependence degree affects the visualization of the data handled by these models, because the number of relationships increases the complexity of the final diagram. Therefore, despite the problems of having a large number of dependencies, the benefit is that, the more dependencies and parameters there are, the more information there is available to extract useful information about the security and QoS trade-off. For this reason future work will aim to provide a recommendation system for extracting information from CPRM-based models where large numbers of parameters coexist.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the project ARES (CSD2007-00004). Additionally, it has been funded by Junta de Andalucia through the projects PISCIS (TIC-6334) and FISICCO (TIC-07223). The author has been funded by the Spanish FPI Research Programme.

References

1. Alia, M., Lacoste, M., He, R., Eliassen, F.: Putting together qos and security in autonomic pervasive systems. In: Proceedings of the 6th ACM Workshop on QoS and Security for Wireless and Mobile Networks, pp. 19–28. ACM (2010)
2. Karatas, F., Kesdogan, D.: A flexible approach for considering interdependent security objectives in service composition. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 1919–1926. ACM (2013)
3. Yu, F.R., Tang, H., Bu, S., Zheng, D.: Security and quality of service (qos) co-design in cooperative mobile ad hoc networks. *EURASIP J. Wirel. Commun. Netw.* **2013**(1), 1–14 (2013)
4. Carapinha, J., Bless, R., Werle, C., Miller, K., Dobrota, V., Rus, A.B., Grob-Lipski, H., Roessler, H.: Quality of service in the future internet. In: Kaleidoscope: Beyond the Internet?-Innovations for Future Networks and Services, 2010 ITU-T, pp. 1–8. IEEE (2010)
5. Ernst, J.B., Kremer, S.C., Rodrigues, J.J.P.C.: A survey of qos/qoe mechanisms in heterogeneous wireless networks. *Phys. Commun.* **13**, 61–72 (2014)
6. Aiash, M., Loo, J.: An integrated authentication and authorization approach for the network of information architecture. *J. Netw. Comput Appl.* **50**, 73–79 (2015). <http://dx.doi.org/10.1016/j.jnca.2014.06.004>
7. Yen, I.-L., Ma, H., Bastani, F.B., Mei, H., et al.: Qos-reconfigurable web services and compositions for high-assurance systems (2008)
8. Golden, B.L., Wasil, E.A., Harker, P.T.: *Analytic Hierarchy Process*. Springer, Heidelberg (2003)
9. Hansen, P., Ombler, F.: A new method for scoring additive multi-attribute value models using pairwise rankings of alternatives. *J. Multi-Criteria Decis. Anal.* **15**(3–4), 87–107 (2008)
10. Nieto, A., Lopez, J.: A context-based parametric relationship model (cprm) to measure the security and qos tradeoff in configurable environments. In: IEEE International Conference on Communications (ICC), pp. 755–760. IEEE (2014)