

Security and QoS Tradeoff Recommendation System (SQT-RS) for Dynamic Assessing CPRM-based Systems

Ana Nieto
Department of Computer Science
University of Malaga
Malaga, Spain
nieto@lcc.uma.es

Javier Lopez
Department of Computer Science
University of Malaga
Malaga, Spain
jlm@lcc.uma.es

ABSTRACT

Context-based Parametric Relationship Models (CPRM) define complex dependencies between different types of parameters. In particular, Security and QoS relationships, that may occur at different levels of abstraction, are easily identified using CPRM. However, the growing number of parameters and relationships, typically due to the heterogeneous scenarios of future networks, increase the complexity of the final diagrams used in the analysis, and makes the current solution for assessing Security and QoS tradeoff (SQT) impractical for untrained users. In this paper, we define a recommendation system based on contextual parametric relationships in accordance with the definition of CPRM. The inputs for the system are generated dynamically based on the context provided by CPRM-based systems.

Categories and Subject Descriptors

C.2 [Computer-Communication Networks]: General

Keywords

QoS, Security, Tradeoff, Context, CPRM, SQT.

1. INTRODUCTION AND BACKGROUND

The concept of *network of the future* is subject to the cooperation and collaboration of entities under the umbrella of improvements, both hardware and software, that enable devices in the network to interact, generating large amount of data. Indeed, this information is invaluable because it is possible to infer information about the user's preferences, network performance, and Quality of Service (QoS). For example, it is usual that security mechanisms such as *Intrusion Detection Systems* (IDS) analyse traffic data stored in data bases by pattern matching to identify threats. Hence, all the data generated as a consequence of the diversity of devices, multiple functionalities, large number of networks and the user's participation in the environment are not discarded.

Instead, new concepts like Big Data emerge as an alternative way for handling large data of different types with great performance. The dependencies between different types of parameters in different contexts, can be extracted from this data using known techniques for data processing such as data mining or, in some cases, the human observation.

Therefore, the networks of the future present the following immediate challenges:

- Heterogeneous networks formed by multiple purpose devices, with mobility capabilities.
- User's participation in future networks.
- The management of large amounts of information, born from the collaboration and convergence of devices, networks and users.

These three aspects are dependent on Security and QoS issues. First, the collaboration between devices and networks is not useful when the performance is damaged. For example, future networks will allow the owner of a home network to share part of its resources with his neighbour, forming a collaboration. However, this is impractical if finally the QoS at home is poor due to this collaboration. In the same way, if the QoS is guaranteed at home but finally the user detects that his neighbour is using his home network beyond the contract agreement, this represents an abuse of trust.

Moreover, security mechanisms for controlling these events could affect the QoS where resource-limited devices are part of the network [1]. It is widely known that a network is only as secure as its weakest part. In this regard, Wireless Sensor Networks (WSNs), formed by sensors and actuators, that in many cases depend on batteries, are exposed to typical threats, like, for example, Denial of Service (DoS). In the past, when WSNs were isolated from the Internet, this problem was avoided, but the new trend is to try to avoid this isolation so as to be able to benefit from the diversity of devices, thereby increasing the functionality.

Furthermore, many security holes in networks start with a poor configuration of network policies, or selection of inadequate services by users. It is common that technology moves too fast for many users. When security mechanisms affect the QoS and vice versa the perception of the user is that the system does not work. Indeed, if new improvements are unable to deal with users, and train them, then the whole system is at risk. For example, one of the scenarios which combines the high convergence and collaboration of multiple devices and users in future networks, is mobile networks, where many of the problems of Security and QoS tradeoff in future internet can be found [2]. In particular,

personal devices add a large degree of uncertainty to the final composition of the network, both software and hardware. Moreover, *malicious* is a terminology directly related with the user's behaviour.

So, the heterogeneity of devices, the user's participation and a large amount of information to be processed and handled, present a risk for the survival of future paradigms and the new technologies that are on the way [3]. In particular, we are concerned about problems that may make the security and QoS inconsistent.

Our approach, takes advantage of the characteristic information from different environments to help decide the final configuration of *things*. From our point of view, the growing diversity of devices and technologies makes assessing the security and QoS tradeoff based on parametric relationships very interesting.

Similar approaches for providing recommendations considering security and QoS are most of them focused on service composition, as is the case of [4] and [5], or in providing recommendations but to be implemented in devices as in [1]. In particular, in [6] a friendly tool to simplify the decisions of the user on the selection of security goals is proposed. However, these approaches are conceived for services, and does not provide dynamic recommendations based on heterogeneous contexts formed by different *things*.

1.1 Context-based Parametric Relationship Models (CPRM)

Context-based Parametric Relationship Models (CPRMs) [7] enable the analysis of different types of parameters (characteristics, properties, etc.) at different abstract layers (user's requirements, measurements, composition of devices) considering *General Contexts* (GC), based on common characteristics between networks, and *Particular Contexts* (PC), which describe specific characteristics of the environment, and may change over time.

The abstract definition of parameters in CPRM-based systems streamlines the analysis of the Security and QoS tradeoff in dynamic-composition networks. This analysis is based on an existing knowledge of the network, that is, information about the use case to be analysed. In our case, which analyses the Security and QoS tradeoff, our CPRM-based system is defined by an extensive set of Security and QoS parameters and their relationships with other types of parameters in order to draw conclusions about the effect that the composition of *things* has on the environment.

In a CPRM-based system, there is no definition based on a physical layered infrastructure. Instead, a parameter is considered as any *thing* that should be analysed, and that depends on or affects other things/parameters. So, the set of parameters and their relationships determine the results of the analysis. And, as it is a knowledge-based system, it is highly dependent on the accuracy of the information. In addition, CPRM-based systems are built according to a set of rules that change the context dynamically, which is very useful for assessing multidisciplinary environments of dynamic composition where it is very difficult to predict with any great accuracy the devices that will form the network.

The problem is that the model becomes more complex the more data that are handled, and the increasing number of parameters and relationships complicate the analysis of results derived from a CPRM-based system. Given the nature of future networks, it is obvious that a CPRM-based system

needs to be enhanced so it can provide recommendations for Security and QoS tradeoff benefitting from the large amount of information available. Similar approaches to provide recommendations like that proposed in [5] for trustworthy web service selection, follow the principle for establishing dependencies between general concepts.

While CPRM-based systems combine *things*, there are alternatives such as the service composition based on interdependencies and QoS constraints [4], and the use knowledge-based systems or ontologies [8, 9]. However, in the end, the previous approaches all concentrate on specific layers (e.g. service), or have been developed with specific purposes in mind (e.g. service oriented architectures).

In this paper, the limitations of parametric-based systems are mitigated. To achieve this, our *Security and QoS Tradeoff* (SQT) tool is improved with a recommendation system (SQT-RS) to help in assessing the Security and QoS tradeoff in scenarios with a large number of parameters, as is the case of future network environments.

1.2 Security and QoS Tradeoff (SQT) tool

SQT implements a CPRM-based systems handler and provides samples of CPRM-based systems, based on a predefined set of parameters and relationships defined at a high-layer of abstraction, focusing on Security and QoS tradeoffs [7]. That means that SQT depends on the set of parameters chosen to operate with the model, and, hence, on the predefined behaviour based on the current literature. This has proved to be useful from the point of view of research, at a high layer, for example, in [7] the use case of instantiation of Authentication mechanisms in WSNs was considered.

However, the current version of SQT can be difficult or impractical for untrained users. The reason for this, is that the final results provided by SQT (graphs) have to be carefully analysed prior to making a decision, and, the growing number of parameters complicates the analysis which, therefore, requires much more time. To make SQT useful for users, it needs to be adapted to provide real-time recommendations based on goals and the current state of the model.

Some improvements are possible by adding new functionality in SQT, in order to set up requirements, and provide recommendations. Specifically, in this paper:

- The concepts of *requirement*, *goal* and *recommendation* for CPRM-based systems are defined.
- *Facts* and *rules* to perform the inference process to identify the best configuration or recommendations given the requirements and goals are defined.
- An example of the generation of facts and recommendations given a predefined set of Security and QoS parameters is discussed.
- The effect of the contextualized parameters on the final number of facts generated by SQT-RS is analysed.

To implement these characteristics, in this paper an expert system based on CLIPS is defined and integrated inside SQT. The information in CPRM-based systems (parameters and relationships), and the results inferred from them (impact and influence of parameters) are processed to produce the facts needed for the expert system to work. The rules are defined and implemented in accordance with the properties of CPRM-based systems.

1.3 Prior formulation

The SQT tool is based on the definition of a CPRM-based system, provided in [7], where a set of operations on the parameters of the model are explained. This formulation is needed to define the recommendation system, because the operations on the parameters help us to determine the final set of recommendations. In the following sections, the terminology in Table 1 will be used.

Table 1: Recursive operations in a CPRM.

Accumulative Influence (ι) and Accumulative Dependence (δ)	
$\iota(a) = I_a , I_a = \{x x \rightarrow a \vee xRa, x \neq a, x \in P\}$	(1)
$\delta(a) = D_a , D_a = \{y a \rightarrow y \vee aRy, y \neq a, y \in P\}$	(2)
$xRy \iff x \rightarrow y \vee \exists k k \in D_x \wedge k \in I_y$	(3)
Impact Increasing (Δ) and Decreasing (∇) a Parameter x	
$\Delta x \implies \forall y xRy, v(y) = v(y) + w_T \wedge u(y, w_T)$	(4)
$\nabla x \implies \forall y xRy, v(y) = v(y) + w_T \wedge u(y, w_T)$	(5)
$u(x, \omega) = \begin{cases} \Delta x & \text{if } \omega > 0; \\ \nabla x & \text{if } \omega < 0; \end{cases}$	(6)

The *Accumulative Influence* (ι) and the *Accumulative Dependence* (δ) are functions applied on a parameter $a \in CPRM$. Both, are based on the cardinality of the sets, respectively, I_a and D_a (Exp. 1-2). I_a groups those parameters that are related with a , being a the consequent in the relationship, while D_a is formed by the parameters that are related to a , being a the antecedent in the relationship. So, these parameters *depend on* the parameter a . In both cases, I and D , the parameters are related directly $x \rightarrow y$, or indirectly through intermediary parameters k (Exp. 3).

Moreover, the impact produced by increasing (Δ) or decreasing (∇) a parameter can be measured based on Exp. 4-5. Note that the impact of the operations (Δ , ∇), will be propagated throughout all the parameters affected by the dependencies. This recursive effect is produced by Exp. 6, based on the value of ω . These steps are explained in more detail in [7]. For what follows, it is enough to understand that w_T depends on the type of relationship defined between the parameters in the dependence (+, -, ...). We will come back to this point when defining the recommendation set.

The rest of the paper is structured as follows. Section 2 defines the concepts of goals/requirements and recommendations subject to a CPRM-based system. Section 3 provides the definition of facts and rules for the inference process, built on the expert system. Finally, in Section 4 an example of generation of facts given a predefined set of Security and QoS parameters is discussed.

2. CPRM-BASED STRUCTURES IN SQT-RS

In this section we present the concepts of goal, requirement and recommendation according with the definition of CPRM, and how they are used in SQT-RS.

2.1 Goals and Requirements

Goals and requirements are organized into two structures GOA and REQ (Exp. 7-10). The first tuple in these structures is for general descriptions. Thus, GOA is described based on the number of goals ($\#G$) included, the identifier of the model for which the goals are defined ($CPRM_{id}$), and the number of recommendations provided ($\#Rec$). After the general description, GOA includes the goals (g_K). The goals are described based on an identifier (id), an

objective parameter given by its identifier (P_{id}), the objective or criterion to be applied, and a list of recommendations to satisfy the goal (S_{id}), that initially is set to null.

$$GOA = \{\{\#G, CPRM_{id}, \#Rec\}, g_1, \dots, g_{\#G}\}; \quad (7)$$

$$g_k = \{id, P_{id}, objective, S_{id}\}; \quad (8)$$

The requirements structure (REQ) includes $\#Req$ requirements for a given $CPRM_{id}$. In this case, recommendations are not applicable, because the requirements will be forced in $CPRM_{id}$. The requirements req_k are described based on an identifier (id), the id of the parameter and the value taken by the parameter (val).

$$REQ = \{\{\#Req, CPRM_{id}\}, req_1, \dots, req_{\#Req}\}; \quad (9)$$

$$req_k = \{id, Parameter_{id}, val\}; \quad (10)$$

Requirements and goals can be added to the system using the GUI shown in Figure 1. The difference between requirements and goals, is that the first define the values that a parameter can take, and the second define objectives in the parameters, therefore, for it to be useful it requires the execution of operations defined in the CPRM.

We consider as objective criterions the maximization (max) or minimization (min) of parameters. Specifically, based on the classification of parameters in CPRM, parameters of type *consequence* are a good candidate to be considered in tuples. For example, (min interference) may be a goal to be considered in the final recommendation.

Moreover, parameters of type *performance* also suit this representation. For example, minimizing the average delay or/and the energy consumption are both topics widely discussed in the current literature. They could be expressed as (min delay) and (min energy_consumption), respectively. When both are given as goals, the final composition will be “(min delay) and (min energy_consumption)”.

To maximize the value of the parameters is also possible. For example, to maximize residual energy (max res – energy) or the network lifetime (max lifetime). In physical layer security, goals based on maximizing the Signal-to-noise ratio (SNR) or the Signal-to-interference ratio (SIR), as well as the min-max rate, are widely discussed.

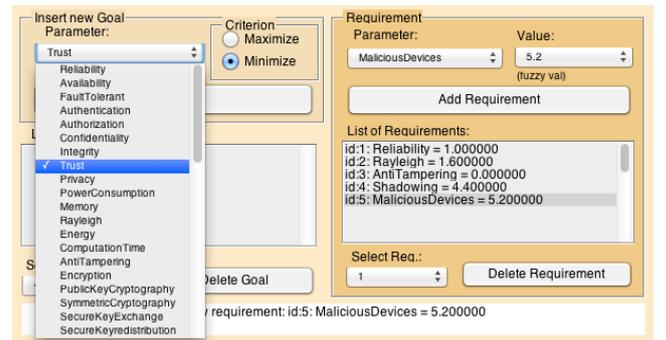


Figure 1: GUI for requirements and goals.

All these restrictions can be set up as a common set of restrictions which will be provided in SQT to infer results.

The final set of objectives will be part of the final set of facts, the inputs for the recommendation system.

2.2 Recommendation

The recommendation is composed by a set of parameters selected to satisfy the objectives or goals introduced by the user in the previous step (max,min). Moreover, given the nature of CPRM, the parameters that are shown depend on the information in the model.

According to the definition of a CPRM-based system, the information in a $CPRM_i$ is richer when more mechanisms are integrated into the model, that is, the more parameters of type *instance* the model has¹. Specifically, if all the parameters are of type *instantiated* or *instance*, the model is completely instantiated, and the final recommendation shows a configuration of services/mechanisms.

A $CPRM_i$ is based on the union of one or various PCs, where the final mechanisms and services are defined. So, the parameters that were instantiated by one or various parameters defined in a PC, lack relevance in the final recommendation, because the system knows the real mechanisms that will be used to implement the parameter.

Consequently, the final recommendation depends on the best configuration of parameters which not only satisfies the list of facts or objectives defined by the user, but also depends on the instance of parameters, let's say, the specific mechanisms available in the system to provide the requirements needed. So, we define some rules that should be considered to provide a recommendation:

- A parameter of type *instantiated* will not be shown. Instead, its instances are shown (RR1).
- Only those parameters which satisfy one or various objectives are shown (RR2).
- The priority of the objectives is based on the identifier, that is assigned in order of when the objective is defined: the first objectives created will have higher priority than the last ones (RR3).
- The list of objectives that cannot be satisfied are clearly identified (RR4).

Note that, based on the aforementioned points, a recommendation is an output of the system, while the goals and facts are inputs. Based on these outputs or recommendations, the user receives feedback, and can change the inputs or the composition of the model. Therefore, the way in which this output is perceived by the user is crucial.

Table 2: Formulation for recommendations.

Generic def. of goal: $g \in \{max, min\}, g :: CPRM \rightarrow [0, 1]$	(11)	
Generic def. of Recommendation: $R = \{id1_{op1}, \dots, idN_{opN}\}$	(12)	
Goal, $g(P)$	Recommendation	
$max(P)$	$R id(x_j) = idj, x_j \in I_P, op_j(x_j) \rightarrow \Delta P$	(13)
$min(P)$	$R id(x_j) = idj, x_j \in I_P, op_j(x_j) \rightarrow \nabla P$	(14)
$\Theta(R) = \sum_{j=1}^N op_{jP}(x_j) idj_{op_j} \in R, op_j \in \{\Delta, \nabla, \square\}$	(15)	
$\square x \Rightarrow \Delta x == \nabla x$	(16)	

The mathematical representation of a recommendation is defined based on a goal or multiple goals. Table 2 shows the formulation considering only one goal and one recommendation set as output. The approach which takes into account

¹ $CPRM_i$, or instantiated CPRM, is a CPRM with a PC integrated. The PC defines mechanisms (instances) for implementing parameters in the original CPRM (instantiated).

several outputs or different recommendation sets can be broken down as follows:

- For a goal g on a parameter P , a recommendation set S , that satisfies a goal, either $S_{max}(P)$ or $S_{min}(P)$, is formed by sets of parameters, denoted as R_i (R in Table 2), ordered depending on the final impact on P , such that $S_g(P) = \{R_1, R_2\} \Leftrightarrow \Theta(R_1) > \Theta(R_2)$.
- op_{jP} means that, from the results of op_j , only the results for P are considered.
- Each parameter x_j in a recommendation set R_i , satisfies that after decreasing, increasing or both, P is enhanced, based on g . Therefore, x_j belongs to the influence set of P (I_P) by definition (Exp. 1).
- Consequently, $R_i = \{id1_{op1}, \dots, idN_{opN}\}$, where $id1 \dots idN$ are identifiers of parameters in CPRM, and op_i is the type of operation through the objective is satisfied.
- \square means that the goal can be achieved by applying either operation Δ or ∇ .

For multiple goals, in the case that a single recommendation set S is unable to satisfy all the objectives, it is possible to provide multiple recommendation sets.

3. FACTS AND RULES FOR THE INFERENCE PROCESS BASED ON CPRM KNOWLEDGE

In the following paragraphs, how to convert the information from the CPRM/ $CPRM_i$ into facts is discussed, and the definition of rules according CPRM is detailed. A general overview of the sequence to build the facts and rules to be used by the expert system is shown in Figure 2.

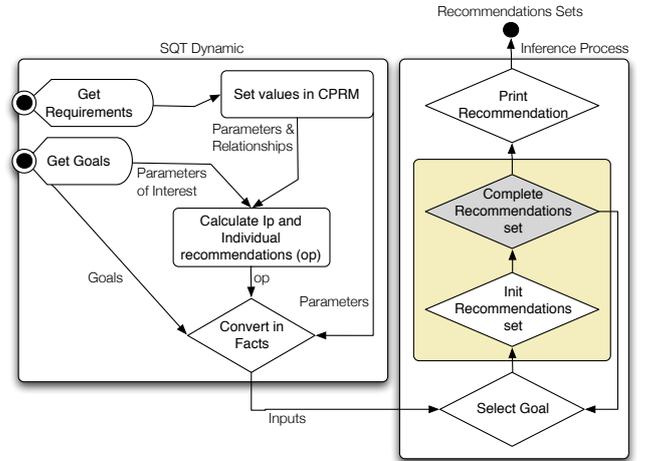


Figure 2: Recommendation chain.

3.1 CPRM-based facts

In a CPRM-based system, the definition of parameters and dependencies is key to extracting information. Intuitively, the information gathered from the model will change according to the values of these parameters and their relationships. Similarly, it is expected that the set of facts will be representative of the current state of the model. In this case, it means that (1) the set of facts have to be generated dynamically, according to the values extracted from

the CPRM selected by the user, while (2) also considering the requirements and goals desired by the user.

Taking into account (1-2), the list of facts provided by SQT-RS to infer information, will be delivered from:

1. The requirements and goals selected by the user: values required and max/min parameters.
2. Current information in the CPRM/ $CPRM_i$: parameters and relationships.

First, the user's requirements are set up in the model, and then, individual recommendations, denoted as op , are calculated based on Exp. 13-14. Note that following this approach, the requirements although not considered as facts, do influence the final recommendation by changing the value of the parameters in the CPRM.

Unlike the requirements, the goals selected by the user are converted to facts without being processed by SQT-RS. The same occurs with the definition of parameters.

In addition, the SQT-RS define the following types of input-facts, considered *dynamic* because they are generated when the preferences/inputs given by the user change the behaviour of a model:

- Goals. Given by the user, they are defined based on a criteria (*max* or *min*) and a parameter.
- Individual recommendations, op . According with Exp.13-14, reflects the operation to be carried out on a parameter in order to satisfy the goal imposed by the user.
- Parameters (information in the model). These can be non-contextual parameters, or parameters with type *instantiated* or *instance* (contextual).

The dynamic input-facts are processed by the static rules defined for SQT-RS, and finally generate output-facts: recommendations based on goals.

Note that according to the mathematical formulation, the individual recommendations are based on the results for the increasing/decreasing of parameters which provides the best result based on the goals. The CPRM defines four types of tests: accumulative dependence, accumulative influence, increasing a parameter, decreasing a parameter. Then, SQT-RS, by using the previous input-facts, will provide recommendations based on the result of these operations, given based on the set of parameters defined in the model and the values to be enhanced as required by the user.

Once the user gets feedback from the tool, new values can be introduced and then the model generates new facts and the inference process starts again.

3.2 CPRM-based rules

The CPRM-based rules are static and never change. They are envisioned to satisfy the requirements detailed in Sections 2.1 and 2.2. So, the inference process is divided into three phases, detailed in Figure 2:

1. Selection of a goal. Repeated while there are goals to be processed.
2. Calculation of the set of recommendations, given the goal. The set of recommendations for a goal has to take into account the type of parameter (contextual or non-contextual).
3. Print results. All the results are printed at the end of the inference process.

While 1 and 3 are basic steps, the greatest processing time is in step 2. In this phase, rules are applied based on the type of parameter to be considered. The simplest rules in this phase are those which consider non-contextual parameters. However, when an individual recommendation is provided by an instantiated parameter, the following property, satisfied by any CPRM-based model, is considered:

Definition (Prop.1): If there is an individual recommendation for an instantiated parameter, that is $\exists op_j(x_j), x_j \in I_P, type(x_j) == instantiated | g(P) = 1$, then, there is a recommendation for each instance of this parameter.

This property is satisfied because the coherence rules in which the construction of any CPRM model is based. Simplifying, as all parameter instances inherit their parent's relationships, then, if $x_j \in I_P, \forall x, y, x \in P(y)$, that is, x belongs to the set of parents of y , then, as y inherits the relationships of x , then if $x \in I_P$, then y belongs to I_P too².

Note that Prop.1 is assumed when the recommendation system is considering an individual recommendation which consists of increasing/decreasing (that is, operating on) an instantiated parameter. This is different from those cases when the instantiated parameter is part of a goal. When the instantiated parameter is part of a goal, Prop.1 is irrelevant. In a CPRM model, all the instantiated parameters inherit the relationships of their instances, when the model required has to be coherent. These new relationships inherited by the instantiated parameters have weight 0 so as not to interfere with the instances (see rules def. in [7]).

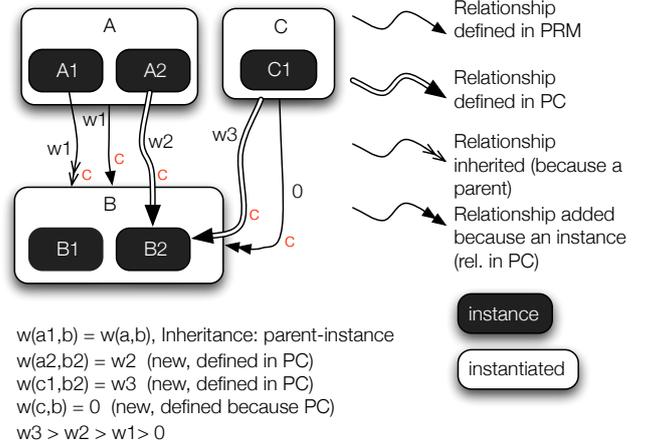


Figure 3: Example of inheritance relationships.

As a result of this, the inference process cannot interpret an instantiated parameter in a goal as a non-instantiated parameter. This is because the weights of the relationships for an instance are more specific than or equal to the weights defined by the parent to the same instance, and different weights provide different individual recommendations. Hence, the expert system, when an instantiated parameter is part of a goal to be satisfied, breaks this goal down into *subgoals* in order to consider any instance of the parameter as a goal to be satisfied, but within the context of maximiz-

²The set of parents of y , $P(y)$, not to be confused with the parameter P , in Prop.1. In this case it is only a variable.

ing/minimizing the instantiated parameter. This is necessary so as to identify the instances that help to maximize the property/parameter. So, the following property is satisfied by SQT-RS:

Definition (Prop.2): A CPRM-based recommendation system considers the problem of maximizing/minimizing an instantiated parameter as the problem of maximizing/minimizing the instances of the parameter. That is: if $\exists g(p) | type(p) == instantiated$, then, $R_p = R_{p_i} | p \in P(p_i)$.

It must be observed, that while Prop.1 is satisfied by the properties of a CPRM, Prop.2, which is implemented in the recommendation system, does not come from the model behaviour. In order to clarify the relevance of this property we use the example depicted in Figure 3. Consider that the goal is in B (*max* or *min*). Then, the inference process will return recommendations about possible modifications on A1, A2 and C1, in order to maximize B. However, without taking into account Prop.2, the final recommendation will consider the weights defined by these parameters with B, and not the weights defined with their instances.

If the goal is, for example, to maximize B, considering only A,B and C, and the relationship between C and B defines a positive impact³ then a good recommendation is to increase or provide the capability C1 in order to maximize the property B through the improvement of B2. In addition, if the goal *max*(B) does not imply *max*(B1), *max*(B2), then, the main recommendation will be to increase A1, and C1 will not be mentioned.

To satisfy Prop.2, the recommendation system must implement, in addition, the property, Prop.3:

Definition (Prop.3): The information about an instance is provided as a fact when any of its parents are considered in a goal or in an individual recommendation. In addition, if the parent is in a goal, then the individual recommendations to the instance are provided too.

This property is required in order to provide the inference process with the information about the instances for creating the subgoals, and inferring information. Independently of Prop.3, the information about an instance can be provided as fact based on the steps detailed in Section 3.1.

The complete set of rules defined for SQT-RS are deployed in Figure 4. A,B and C represent different states related to parameters: non-instantiated and not-instance (A), instantiated (B) and instances (C). In the building of the recommendation phase, when the parameter is instantiated, their instances are considered, and for this reason there are no specific subrules for this state. The subrules r31 and r32 of type B, manage objectives for instantiated parameters. Hence, in these cases, the final recommendation is based on maximizing/minimizing the instances of the instantiated parameter. These rules work under the assumption that Prop.2 and Prop.3 are satisfied. In other words, for any goal defined for an instantiated parameter, there are facts defining the subgoals of the goal, and these subgoals are defined for

³The final impact depends on the symbol of the relationship. In this example, all the relationships are complete (c). That means that when the antecedent increases, the consequent increases, and when the antecedent decreases, the consequent decreases.

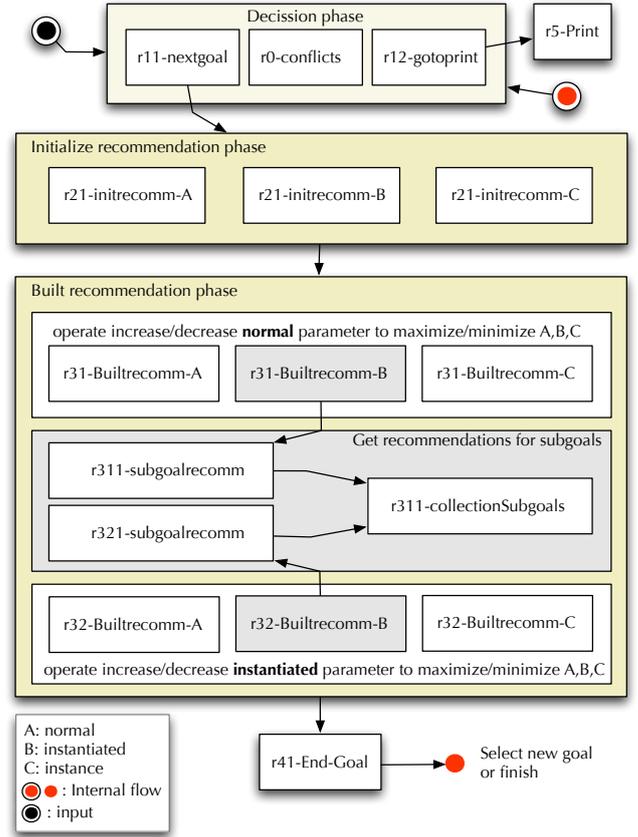


Figure 4: CLIPS rules and phases.

the instances of the instantiated parameter. In addition, the individual recommendations for subgoals are also added as facts too, based on Prop.3.

Finally, the rules are taken directly from a .clp file, and this file is completed with the dynamic facts generated by SQT-RS. For this reason, SQT-RS uses a .jar file, developed to assert, dynamically, the facts provided by SQT-RS, in a CLIPS environment where the rules and templates are defined using Jess [10]. The recommendation sets are stored in a temporary file and processed by SQT-RS in order to show the results to the final user. Moreover, the conflicts that avoid the satisfaction of multiple goals are collected by the rule r0-conflicts and stored in an additional file that is shown to the user, using SQT-RS.

4. ANALYSIS

In this section we provide some results of SQT-RS based on the goal maximize Energy (*max Energy*), using the PC analysed in [7], which instantiates the parameter Authentication, in a Wireless Sensor Network (WSN) scenario.

Figure 5(a) shows the file of facts generated in this example by SQT-RS. We focus in the goal *max Energy* because the file of facts generated is smaller compared with the goal *min Energy*. This is because of the types of relationships defined in the PRM source. We also select Energy because Authentication, which is an instantiated parameter, affects it. So, the file of facts also shows the facts generated for Authentication and the instances of this parameter. Moreover,

```

(goal (priority 1) (criterion maximize) (parameter 13))
(op (todo decrease) (on 4) (to maximize) (p 13) (val 66.000000))
(op (todo decrease) (on 10) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 12) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 23) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 24) (to maximize) (p 13) (val 1.000000))
(op (todo increase) (on 25) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 26) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 27) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 28) (to maximize) (p 13) (val 1.000000))
(op (todo increase) (on 29) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 30) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 32) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 36) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 44) (to maximize) (p 13) (val 1.000000))
(op (todo decrease) (on 49) (to maximize) (p 13) (val 102.000000))
(op (todo decrease) (on 50) (to maximize) (p 13) (val 106.000000))
(op (todo decrease) (on 51) (to maximize) (p 13) (val 38.000000))
(op (todo decrease) (on 52) (to maximize) (p 13) (val 66.000000))
(parameter-instantiated (id 4) (name "Authentication") (layerP "High-Layer") (typeP "Security"))
(parameter (id 10) (name "PowerConsumption") (layerP "LocalProperties") (typeP "Performance"))
(parameter (id 12) (name "Rayleigh") (layerP "LocalProperties") (typeP "Performance"))
(parameter (id 13) (name "Energy") (layerP "LocalProperties") (typeP "Performance"))
(parameter (id 23) (name "DataRate") (layerP "Communication") (typeP "Performance"))
(parameter (id 24) (name "PacketSize") (layerP "Communication") (typeP "Performance"))
(parameter (id 25) (name "SignalStrength") (layerP "Communication") (typeP "Performance"))
(parameter (id 26) (name "DataTransmission") (layerP "Communication") (typeP "Performance"))
(parameter (id 27) (name "TransmissionTime") (layerP "Communication") (typeP "Performance"))
(parameter (id 28) (name "TransmissionPower") (layerP "Communication") (typeP "Performance"))
(parameter (id 29) (name "TimeSleeping") (layerP "Communication") (typeP "Characteristics"))
(parameter (id 30) (name "RequiredTimeOn") (layerP "Communication") (typeP "Characteristics"))
(parameter (id 32) (name "Retransmission") (layerP "Communication") (typeP "Consequences"))
(parameter (id 36) (name "PacketLoss") (layerP "Measurements") (typeP "Performance"))
(parameter (id 44) (name "Congestion") (layerP "Environment") (typeP "Consequences"))
(parameter-instance (id 49) (name "CAS") (layerP "High-Layer") (typeP "Security") (id-parent 4))
(parameter-instance (id 50) (name "DAS") (layerP "High-Layer") (typeP "Security") (id-parent 4))
(parameter-instance (id 51) (name "IDS") (layerP "High-Layer") (typeP "Security") (id-parent 4))
(parameter-instance (id 52) (name "IMBAS") (layerP "High-Layer") (typeP "Security") (id-parent 4))

```

To maximize Energy:
-> decrease DAS(ef:106.0);
-> decrease CAS(ef:102.0);
-> decrease IMBAS(ef:66.0);
-> decrease IDS(ef:38.0);
-> decrease Congestion(ef:1.0);
-> decrease PacketLoss(ef:1.0);
-> decrease Retransmission(ef:1.0);
-> decrease RequiredTimeOn(ef:1.0);
-> increase TimeSleeping(ef:1.0);
-> decrease TransmissionPower(ef:1.0);
-> decrease TransmissionTime(ef:1.0);
-> decrease DataTransmission(ef:1.0);
-> increase SignalStrength(ef:1.0);
-> decrease PacketSize(ef:1.0);
-> decrease DataRate(ef:1.0);
-> decrease Rayleigh(ef:1.0);
-> decrease PowerConsumption(ef:1.0);

(a) Facts for maximizing Energy. (b) Recommendation set.

Figure 5: Example.

the facts are shown only for testing purposes. The aim, is that this file will only be used by SQT-RS, the final user can only see the final recommendations. The recommendations set given the facts in Figure 5(a), are shown in Figure 5(b).

In Figure 5(a) only Authentication is instantiated. Specifically, CAS, DAD, IDS and IMBAS are instances of Authentication. The rest of the parameters remain within the default value of effect 1, while the effect for increasing/decreasing the instances is conditioned to the values given in the PC. In addition, the final recommendations set, does not consider instantiated parameters. Instead, it takes the results given by the instances of the parameters. The results shown in Figure 5(b) first shows the individual recommendations which affect the goal to a greater extent, given the context.

For example, given the first individual recommendation, it is possible to estimate that, for maximizing the parameter Energy, if we are providing Authentication mechanisms, we should choose IDS instead of DAS. Because if DAS is avoided, the effect (ef) on Energy is reduced by 106. This effect is the result of increasing and decreasing the parameters in the individual recommendations.

Note that Energy is a non-instance, non-instantiated parameter. The rules for building the recommendation are different from the contextual cases. Therefore, the selection of Energy as a parameter in the goal is only complicated by the number of relationships which affect Energy.

Intuitively, the presentation of the results can be enhanced depending on the user. We have chosen the representation in Figure 5(b) because we think that it is very intuitive given the formulation of the problem in this paper. However, this text was generated by the rules in CLIPS, and can be modified, if required.

Furthermore, equally it is possible to provide additional feedback to SQT-RS from the .clp file, taking advantage

of the powerful interpretation of Matlab of strings as commands, functions, etc. It is not complicated to provide automatic feedback to SQT-RS, simply by using the structures according to Section 2. Said structures can be interpreted from SQT-RS and set up in the CPRM chosen.

Finally, the file in Figure 5(a) has a reduced set of parameters. This is because SQT-RS only converts to facts, those parameters that are part of a set of interest. The set of interest is formed by any parameter in a goal, and, if the parameter in the goal is of type instantiated, then, their instances are added as subgoals. After that, the individual recommendations for the parameters in the set of interest are retrieved. This criterion of selection of information, based on the set of interest, is to make it more efficient. Indeed, it is possible to enhance the rules for inferring additional information based on the whole set of parameters, if the conversion to facts is not restricted just to the parameters in the set of interest. However, the size of the files of facts can grow too much in these cases. In the current version of SQT-RS, the size of the file of facts depends on the number of parameters in goals, the number of parameters in goals that are instantiated, and the accumulative influence degree of the parameters in the set of interest. If all the parameters are selected, the number of facts in the file of facts, based on the number of parameters, and considering the additional facts produced by instantiated parameters, is defined as follows:

$$\#Facts = \sum_{j=1}^M (i(x_j) + 1)(l^i(x_j) + 2) \quad (17)$$

Where:

M : is the number total of parameters.

$i(x)$: is a function which returns the number of instances of a parameter x : $|H_x|$, $H_x = \{y|x \in P(y)\}$.

$\iota(x)$: is the accumulative influence of the parameter x , as defined in Table 1.

$\iota^i(x)$: considers the accumulative influence when the parameter x is instantiated.

$$\iota^i(x) = \begin{cases} \sum_{y \in H_x} \iota(y) & \text{if } \text{type}(x) \text{ is "instantiated"} \\ \iota(x) & \text{in other case} \end{cases} \quad (18)$$

Considering E , N and L , respectively, the total number of non-contextual parameters, instances and instantiated parameters, such that $M = E + N + L$. The number of facts based on the contextual number of parameters is shown in Figure 6. The average of instances per parameter instantiated can be calculated as:

$$\bar{i} = 1/L \left(\sum_{i=1}^L |\{y | x_i \in P(y)\}| \right) \quad (19)$$

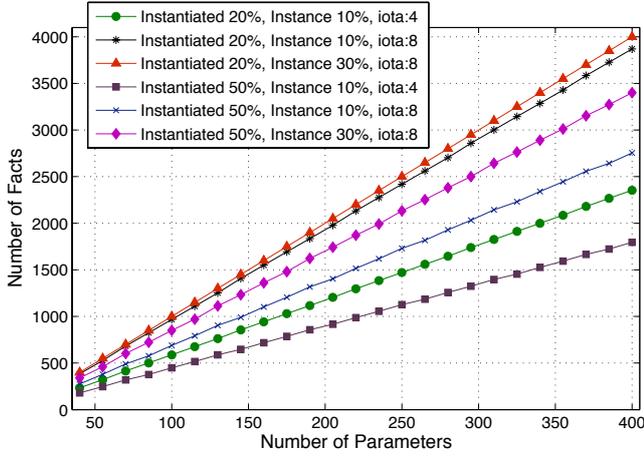


Figure 6: Increasing facts based on the context.

Figure 6 shows the effect that increasing the number of instantiated parameters, instances and accumulative influence (ι , ι) has on the set of facts. As can be seen, the worst cases are registered when the accumulative influence increases. Concretely, the worst case is when ι and the percentage of instances increases, followed by the increase of instantiated parameters. So, the number of instances per instantiated parameter and the accumulative influence are key factors which affect the size of our files of facts.

Moreover, the values chosen for ι in Figure 6 are fixed, while it is obvious that ι varies according to each parameter in the model. Specifically, the parameters which provide a property, such as *Authorization*, should have lower ι than the parameters which define a resource, such as *Memory* or *Energy*. While higher values of ι increase the complexity and the information to be stored and processed, a low value reduces the number of recommendations provided by SQT-RS for those parameter.

5. CONCLUSIONS

The boundary of CPRM-based systems for measuring the Security and QoS tradeoff has been tested in previous work. However, CPRM-based systems are dependent on the number of parameters. When the number of parameters increases, the final results are very difficult to analyse for

a human. In this paper we have overcome this limitation by defining a recommendation system, named SQT-RS, to advise the final user on the alternatives for characteristics, properties and mechanisms to be deployed. SQT-RS can be used with different types of parameters, not only Security and QoS. However, the basic set of parameters provided by the tool defines these types of relationships and no others. These can then be added dynamically by modifying the files generated by SQT-RS. Furthermore, SQT-RS can be particularly useful for assessing Security and QoS tradeoffs in dynamic networks, where there is a great uncertainty about the final composition of mechanisms, services, applications and multipurpose entities.

Acknowledgments

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

6. REFERENCES

- [1] F R. Yu, H. Tang, S. Bu, and D. Zheng. Security and quality of service (qos) co-design in cooperative mobile ad hoc networks. *EURASIP Journal on Wireless Communications and Networking*, 2013(1):1–14, 2013.
- [2] M. Aiash, G. Mapp, A. Lasebae, et al. Integrating mobility, quality-of-service and security in future mobile networks. In *Electrical Engineering and Intelligent Systems*, pages 195–206. Springer, 2013.
- [3] T. Taleb, Y. H. Aoul, and A. Benslimane. Integrating security with qos in next generation networks. In *Global Telecommunications Conference (GLOBECOM 2010)*, 2010 IEEE, pages 1–5. IEEE, 2010.
- [4] T. Yu and K. Lin. Service selection algorithms for composing complex services with multiple qos constraints. In *Service-Oriented Computing-ICSOC 2005*, pages 130–143. Springer, 2005.
- [5] M. Mehdi, N. Bouguila, and J. Bentahar. Probabilistic approach for qos-aware recommender system for trustworthy web service selection. *Applied Intelligence*, pages 1–22, 2014.
- [6] F. Karatas, M. Bourimi, and D. Kesdogan. Towards visual configuration support for interdependent security goals. In *Online Communities and Social Computing*, pages 375–384. Springer, 2013.
- [7] A. Nieto and J. Lopez. A context-based parametric relationship model (cprm) to measure the security and qos tradeoff in configurable environments. In *IEEE International Conference on Communications (ICC)*, pages 755–760. IEEE, 2014.
- [8] J. Carapinha, R. Bless, C. Werle, et al. Quality of service in the future internet. In *Kaleidoscope: Beyond the Internet?—Innovations for Future Networks and Services, 2010 ITU-T*, pages 1–8. IEEE, 2010.
- [9] H. Mostafa, N. Soule, N. Hoff, P. Pal, et al. Applying distributed optimization for qos-security tradeoff in a distributed information system. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1261–1262, 2013.
- [10] E. Friedman-Hill. *JESS in Action*. Manning Greenwich, CT, 2003.