Metrics for QoS analysis in dynamic, evolving and heterogeneous CONNECTed systems

Antinisca Di Marco, Antonia Bertolino, Felicita Di Giandomenico, Paolo Masci, Antonino Sabetta

Metrics for QoS analysis in dynamic, evolving and heterogeneous CONNECTed systems

Antinisca Di Marco
Dipartimento di Informatica
Università dell'Aquila, Italy
antinisca.dimarco@univaq.it

Antonia Bertolino
Felicita Di Giandomenico
Paolo Masci
Antonino Sabetta
CNR-ISTI, Pisa, Italy
name.surname@isti.cnr.it

ABSTRACT

Dynamic, evolving systems pose new challenges from the point of view of Quality of Service (QoS) analysis, calling for techniques able to combine traditional offline methods with new ones applied at run-time. Tracking the evolution and updating the assessment consistently with such system evolution require not only advanced analysis methods, but also appropriate metrics well representative of QoS properties in the addressed context. The ongoing European project CONNECT addresses systems evolution, and aims at bridging technological gaps arising from heterogeneity of networked systems, by synthesising on-the-fly interoperability connectors. Moving from such ambitious goal, in this paper we present a metrics framework, whereby classical dependability/QoS metrics can be refined and combined to characterise CONNECT applications and to support their monitoring and analysis.

Categories and Subject Descriptors

C.4.6 [**Computer Systems Organization**]: Performance of systems—*reliability, availability, and serviceability*; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

1. INTRODUCTION

Our everyday activities and society welfare are increasingly reliant upon the assistance of pervasive inter-connected digital systems. The effective interoperability between such networked systems, which are the most heterogeneous, is strongly technology-dependent and is jeopardised by the fast pace at which technology evolves.

The European Project CONNECT[1] aims at dropping the heterogeneity barriers that prevent networked systems from being *eternally* CONNECTed and at enabling their seamless composition in spite of technology diversity and evolution.

To overcome the dependence upon technology in CONNECT we make minimal assumptions about the networked systems. We assume that they just expose minimal (semantic) description about their functional and non-functional aspects. Building on discovery and learning, CONNECT targets the dynamic synthesis of CONNECTors via which networked systems communicate. The resulting emergent CONNECTors then compose and further adapt the interaction protocols run by the CONNECTed systems (see Figure 1).

The communication between two heterogeneous components speaking different languages can only be achieved by sharing a communication protocol and establishing a common semantics behind the different spoken terms. These two ingredients (communication protocol and semantics) will form the two basic building blocks of the synthesised CONNECTor, for ensuring functional compliance in the inter-components communication. They are however necessary prerequisite, but not sufficient *per se* to guarantee that the CONNECTed networked systems will properly cooperate. To achieve effective communication it is also necessary to provide guarantees of Quality of Service and other non functional properties, such as reliability (e.g., the CONNECTor will ensure continued communication without interruption), security and privacy (e.g., transactions do not disclose confidential data), trust (e.g., networked systems are put in communication only with parties they trust).

The dynamic and evolving context of CONNECT calls for enhanced methods for dependability/QoS-related analysis, which combine traditional offline methods with new ones applied at run-time. In our view, this symmetrically leads to new challenges also from the point of view of defining appropriate metrics which drive the analysis process. In fact, in dynamically evolving systems the dependability properties to be analysed cannot be simplistically derived from the existing taxonomies, like in [1], but may themselves evolve and adapt to the system specific needs.

Therefore, soon after project start, we realised that the CONNECT vision opens to the need of possibly new and more complex metrics for dependability/QoS-related analysis. The

[1]CONNECT : Emergent Connectors for Eternal Software Intensive Networked Systems (http://www.connect-forever.eu) is an FP7 FET Proactive IP Project, G.No.231167, addressing the Theme: ICT Forever Yours

WODA 2010 Trento, Italy

above examples of non-functional properties in fact need to be expressed using suitable metrics that can soon become very complex (as they could mix in different ways several metrics belonging to different domains, such as reliability, performance, security, etc.) and that cannot be defined a priori. Indeed, in CONNECT we cannot know who will ask for a CONNECTor, the language they will use to express the non-functional properties of the request, and who will be the target of the CONNECTion. The investigation of whether and how we could adopt existing metrics, or rather needed to devise new ad hoc metrics, has eventually convinced us of the opportunity to define a new metric classification framework characterised by orthogonal dimensions, as we explain in the following.

In this work, we introduce the developed framework, along which we revisit classical dependability/QoS-related metrics to account for evolution and dynamic interactions in open, evolutionary networks. Relevant properties to be ensured can be elicited from this framework and constitute the reference basis for static and dynamic dependability analyses.

The presentation proceeds as follows: Section 2 presents the challenges of the CONNECT project; Section 3 reports the motivation of our work; Section 4 presents the metric framework devised for CONNECT project, and Section 5 outlines an example of application. Finally, Section 6 hints at the ongoing formalisation of the metric framework and Section 7 concludes the paper.

2. CONNECT CHALLENGES

We depict schematically in Figure 1 the architectural vision of CONNECT. Four types of entities populate the CONNECT world: the enablers, the CONNECTors synthesised by the enablers, the networked systems that operate in the CONNECT world, and the CONNECTed system that is obtained by CONNECTing the networked systems.

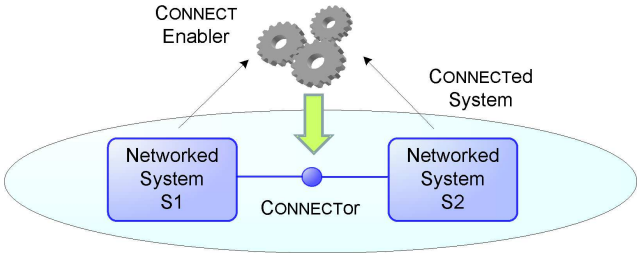


Figure 1: CONNECT vision

Enablers represent the core of CONNECT: they can accept requests from networked systems, discover new networked systems, gather / learn information on their functional and non-functional behaviour, and synthesise a suitable CONNECTor that allows inter-operation among networked systems willing to interact. Thus, we have different types of enablers, according to their provided functionality: discovery enablers, learning enablers, synthesis enablers, monitoring enablers, and so on.

Synthesised CONNECTors are concrete emergent system entities that are dependable, unobtrusive, and evolvable, while not compromising the quality of software applications. To reach these objectives, the CONNECT project undertakes interdisciplinary research in the areas of behaviour learning, formal methods, semantic services, software engineering, dependability, and middleware. More specifically, CONNECT research covers the following issues and related challenges:

- *Discovery* - using behaviour and Quality of Service (QoS) information, rather than just syntactic information. We will employ 'semantic specifications' based on ontologies, which allow an easy symbolic treatment of compatibility from an abstract service perspective.
- *Learning* - dynamically inferring specifications from the networking/communicating behaviour of digital systems; this includes the functional aspects, which are needed to learn CONNECTor behaviours, and the non-functional aspects, i.e., the expectations about QoS and dependability that networked systems have on communication.
- *Synthesis* - synthesising and implementing the specifications of CONNECTors bridging interacting digital systems.
- *Dependability analysis and assurance* - two complementary issues need to be addressed: (i) verification and validation techniques to ensure that networked systems as well as the generated bridging CONNECTors behave as specified with respect to functional and non-functional properties, and (ii) providing adequate security, trust and privacy assurance for interacting parties.

Among the above challenges, this paper specifically focuses on the last item, about which we present the dependability and QoS-related metrics framework we intend to adopt to perform offline and online analysis in CONNECT.

3. MOTIVATIONS FOR A CONNECT METRICS FRAMEWORK

The final aim of CONNECT is to synthesise on-the-fly CONNECTors to allow the communication between two heterogeneous networked systems speaking different languages. One of the challenges here is that the languages, and the relative semantics, are (possibly) unknown to the CONNECT infrastructure when the connection request is issued by a networked system.

The connection request contains functional as well as non-functional requirements that must be analysed and understood by the CONNECT Enablers. The communication can be successfully established only if both functional and non-functional requirements can be satisfied.

From the non-functional point of view, there are many points to be addressed:

- non-functional requirements specification: how does the CONNECTed system specify its requirements on the connection?

- interpretation of requirements and the related metrics: how does the CONNECT Enabler understand the requirements and the metrics used to express it? The non-functional requirements are expressed on the basis of a metric. In CONNECT, new metrics might be necessary. Indeed, CONNECT metrics could be very complex since they can be defined by combining several (classical) metrics. Moreover, an additional problem should be addressed since the metrics could be defined by using the terminology coming from the application domain of the networked system(s) that, in general, may change over time.

- model definition: depending on the metrics and the requirements, suitable analysis models must be defined. Such models are used to give a quantitative assessment to the metrics, and to improve the CONNECTors in case the requirements are not satisfied or in case the evolution of the systems or their contexts induces a degradation of the non-functional characteristics.

- on-line and off-line analysis techniques: the online analysis is based on monitoring techniques that capture the run-time system characteristics indicative of a non-functional problem; off-line analysis instead uses complex models fed by data observed by the monitor to predict worrying trend that could bring the system towards the unfulfilled requirements.

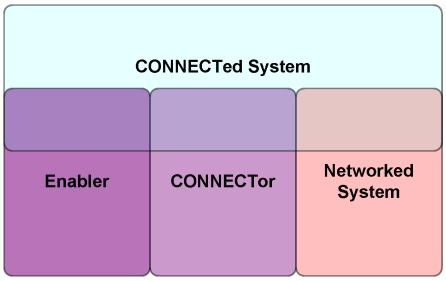
All the above points have the metrics as common ground. The framework we propose revisits the classical dependability and QoS-related metrics to account for evolution and dynamic interactions in open, evolutionary networks. Then, relevant properties to be ensured in the specific system/application at hand can be elicited from this framework in such a way that then it is understandable to CONNECT Enabler.

Therefore, the proposed framework contributes to dynamic-analysis techniques by providing the reference basis for the metrics to be assessed through the analysis. By driving the overall analysis process, this is a necessary, initial step to accomplish both static and dynamic analysis.

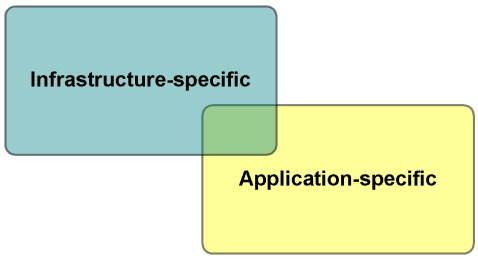
Finally, the classification provided by the framework helps to devise a generative procedure of monitors that can be a composition of different more simple ones.

4. CONNECT METRICS FRAMEWORK

Metrics are quantitative / qualitative indicators suitable to point out the ability of a system to accomplish its intended service according to its specification or user requirements. Classic QoS-related metrics [1] of computer systems, such as dependability, performance, security and trust, need to be conveyed in the CONNECTed world, where the QoS concept may change in accordance with the evolution of the environment. Indeed, new networked systems may join or leave the network, possibly in unforeseen or even unpredictable manners. Hence, the distinction between a *normal* state of the system, and an *error* state from which the system needs to be recovered becomes less neat and the notion of dependability evolves accordingly, as elaborated in [4].



(a) CONNECT-dependent refinement



(b) Context-dependent refinement

Figure 2: Dimensions to derive CONNECT metrics.

CONNECT enablers may need to synthesise CONNECTors on-the-fly, even when the knowledge on the behaviour and capabilities of some networked systems is still incomplete. In these cases, CONNECT enablers may initially synthesise a basic CONNECTor that permits only some elementary form of interaction, and an enhanced CONNECTor may be synthesised only in a second phase, when CONNECT enablers have learnt the behaviour of the new networked systems. Because of the on-the-fly CONNECTor synthesis, it may happen that new networked systems may not be able to get access to a service that is already available in the network, e.g., because CONNECT enablers take some time to synthesise a proper CONNECTor. In certain cases, some networked systems might never be able to get a service, e.g., because of critical functional or non-functional mismatches that cannot be properly bridged by the enablers with the resources available in the network. Similarly, networked systems of an already CONNECTed system may experience service discontinuity or different levels of QoS during time since different CONNECTors might be used in different time frames.

CONNECT metrics need to take into account the above mentioned factors. In order to specify CONNECT metrics, we define a conceptual framework that refines classical dependability metrics along a CONNECT-*dependent* dimension and a *context-dependent* dimension. In the following section, we elaborate the two refinement dimensions and show how they can be used to customise classical dependability metrics with respect to the four actors of the CONNECT architecture (see Figure 1), to the application scenario, and to the heterogeneous / evolvable aspects of the actors of CONNECT.

4.1 CONNECT-dependent dimension

The CONNECT-*dependent* dimension refines generic metrics according to the structural roles of the CONNECT architecture. The rationale behind this dimension is that different definitions of classical dependability metrics may be given

for the different actors of the CONNECT architecture, namely the Networked Systems, the Enablers, the CONNECTors and the CONNECTed System.

With reference to the above actors, the CONNECT-dependent dimension includes three disjoint classes: *NetworkedSystem-specific*, *Enabler-specific*, CONNECT*or-specific*, plus a fourth partially overlapping class, CONNECT*edSystem-specific* (see Figure 2(a)). The *Enabler-specific* and CONNECT*or-specific* classes can be used to obtain "internal" CONNECT metrics, i.e., metrics suitable to assess the dependability level of the CONNECT service.

4.2 Context-dependent dimension

The *context-dependent* dimension refines and classifies generic metrics according to the application context. The rationale behind the choice of this dimension is that CONNECT metrics can be linked to a particular application scenario and / or to heterogeneous and evolutionary aspects of the different actors of the CONNECT architecture. Indeed, as reported in Deliverable D1.1, CONNECT Enablers can accept CONNECT requests from different Networked Systems, discover new Networked Systems, gather / learn information on functional and non-functional behaviour of new Networked Systems, and synthesise, at run-time, new CONNECTors which allow inter-operation among Networked Systems willing to interact.

According to the CONNECT vision, the context-dependent dimension includes two partially overlapping classes (see Figure 2(b)): *application-specific*, which refines generic metrics on the basis of the application domain, e.g., safety-critical, delay-tolerant, real-time; *infrastructure-specific*, which refines generic metrics according to heterogeneity and evolution capabilities of the different actors of the CONNECT architecture, e.g., timeouts adopted by communication protocols, number of operational phases.

5. EXAMPLE OF CONNECT METRICS

In this section, as an exercise of application of the conceptual metrics framework described before, we derive and classify CONNECT metrics for the "Stadium Warning System" scenario, which is one of the CONNECT scenarios [2].

The Stadium Warning System scenario deals with a service used in emergency situations. The goal of the warning system is to dispatch, in the case of danger, an alert message to all people in the stadium. People are assumed to have smart-phones, and the alert message signals a risk and provides instructions to guide people to proper exits. This scenario has been selected since it can be used to emphasise a number of dependability aspects that are of special interest in CONNECT.

The example is elaborated as follows: (i) we start from a set of generic metrics of interest that are relevant to the Stadium Warning System; (ii) we derive CONNECT metrics by refining the selected generic metrics along the different dimensions of the conceptual metrics framework. The selection of generic metrics used in the example, as well as of the refinements that we apply, are made just as a demonstration of the framework, and not as the definitive metrics or relevance for the Stadium Warning System.

Selected generic metrics

We consider the following four generic metrics:

Coverage Probability (Generic Reliability Metric)
Probability for a system to deliver a service to a certain percentage of users.

Latency (Generic Performance Metric)
Maximum/minimum/average expected delay incurred in communicating a message.

Timely Coverage Probability (Generic Performability Metric)
Probability for a system to deliver a service to a certain percentage of users in T seconds.

Confidence (Generic Trust Metric)
Probability that a trusted system performs a particular action.

Derived CONNECT metrics

For each of the above generic metrics, we consider a class along each of the dimensions CONNECT-*dependent* or *context-dependent*, and within it we instantiate the generic metric into a more concrete metric. In the examples shown below, we construct a tag that specifies which class has been considered among those shown in Figure 2. For instance, the first set of metrics refined from coverage probability is collected below the tag: **Connect*edSystem-specific*, Application-specific**, which means that the system to which we refer is the CONNECTed System, and the service and users referred to in the definition of coverage probability are instantiated to the specific services and users of the application of interest, in this case the warning system. Note that in some case a tag includes only one class (i.e., we only refine the generic metric along one dimension), or three classes (this means that along the *context-dependent* dimension we put ourselves in the overlap of the two classes: *Application-specific* and *Infrastructure-specific*).

Metrics derived from Coverage Probability

Connect*edSystem-specific, Application-specific*

- Probability that a certain percentage of smartphones display the same alert message when located in the same area
- Probability to deliver an alert message to all people in the stadium
- Probability to deliver an alert message to all people in the stadium when adversaries are present (assuming that adversaries are able to jam a certain number of messages)

Enabler-specific, Application-specific

- Probability of successful synthesis of a CONNECTor which allows to deliver an alert message to a certain percentage of people in the stadium
- Probability to improve the delivery ratio of a CONNECTor which allows to deliver an alert message to a certain percentage of people in the stadium

NetworkedSystem-specific, Application-specific, Infrastructure-specific

- Probability to successfully display an alert message on smartphones brand X
- Probability that a smartphone brand X receives duplicates of the same alert message

Connect*edSystem-specific, Application-specific, Infrastructure-specific*

- Probability to display alert message A on a set of heterogeneous smartphones located in the same area

Metrics derived from Latency

Connect*or-specific*

- Time to deliver a message from n providers to m users

Enabler-specific

- Time to synthesise a CONNECTor between n providers to m users

Connect*edSystem-specific, Enabler-specific*

- Probability to synthesise a new CONNECTor which reduces message delivery time of d seconds

NetworkedSystem-specific, Infrastructure-specific

- Time to display a received message on smartphones brand X

Connect*edSystem-specific*

- Time to deliver a message to a given percentage of people which move at average speed X

Connect*edSystem-specific, Application-specific*

- Time to complete the registration process of a smartphone to the warning system
- Time to deliver an alert message to a given percentage of people located in the stadium

Connect*edSystem-specific, Application-specific, Infrastructure-specific*

- Time to deliver an alert message to a set of people with a certain percentage of heterogeneous smart-phones

Metrics derived from Timely Coverage Probability

Connect*edSystem-specific, Application-specific*

- Probability that a certain percentage of smartphones receive an alert message in T seconds
- Probability to deliver an alert message in T seconds to all people in the stadium when adversaries are present (assuming that adversaries are able to jam a certain number of messages)

Enabler-specific, Application-specific

- Probability of successful synthesis of a CONNECTor which allows to deliver an alert message to a certain percentage of smartphones in T seconds

NetworkedSystem-specific, Infrastructure-specific

- Probability to successfully display a message in T seconds on smartphones brand X

Connect*edSystem-specific, Application-specific, Infrastructure-specific*

- Probability to display the same message in T seconds on a set of heterogeneous smartphones located in the same area

Metrics derived from Confidence

Connect*edSystem-specific, Application-specific*

- Probability that a certain percentage of trusted neighbours re-broadcast an alert message

Connect*or-specific*

- Probability that a trusted CONNECTor allows communication with a certain percentage of trusted Networked Systems

Enabler-specific

- Probability that a trusted Enabler is able to synthesise a proper CONNECTor

NetworkedSystem-specific, Application-specific, Infrastructure-specific

- Probability that trusted smartphones brand X re-broadcast an alert message
- Probability that a certain percentage of mobile trusted neighbours re-broadcast an alert message

6. TOWARDS AN IMPLEMENTATION OF THE FRAMEWORK

As of this writing, a reference implementation of the metrics framework is being worked on. The goal of this implementation is to assess the usefulness of the framework and its applicability to concrete real-life scenarios. As a preliminary step of this endeavour, we are elaborating on the current specification of the framework in order to define a metamodel to support metrics definition and representation. In the Model-Driven Engineering community [5], *metamodels* "define the relationships among concepts in a domain and precisely specify the key semantics and constraints associated with these domain concepts". In our case, we use a metamodel as a language to specify QoS metrics in CONNECT; semantic constraints can be expressed at the metamodel level. A metamodel not only enables a more precise specification of metrics, but also allows us to express the metrics in a machine-processable format, fostering interoperability with other systems and paving the way to integration in a self-managed control loop.

In practice, metrics models will be expressed in an XML-based format (XMI) which will be the common interchange format among the CONNECT modules (Enablers, Connectors, Networked Systems, etc.). Model-driven technology will be employed (e.g., automated model-transformation, code generation) to manipulate these models and to convert them into other representations, that are readily usable, e.g., for analysis purposes or to automatically generate the code of monitors.

7. CONCLUSIONS

We have briefly introduced the European Project CONNECT, within which dynamic analysis of non-functional properties is a strong concern, due to its runtime derivation of CONNECTors through which networked systems interoperate. In this paper we have focused on the QoS and dependability metrics framework that is currently under development [3]. The aim of the framework is twofold. On the one hand, it provides a way to derive complex refined metrics that are of interest for the analysis of the CONNECT application scenarios. On the other hand, the classification helps to devise a composite procedure to the generation of monitors and to support Verification and Validation activities, through combination of static and dynamic approaches.

CONNECT works under the assumption that the networked systems expose, *a priori*, minimal (semantic) descriptions about their functional and non-functional aspects, and about the way they communicate. This assumption implies that the CONNECTor synthesis must be performed being aware of the metrics coming from the networked systems, which may require a "learning step".

CONNECT must guarantee dependability, security, and trust properties to the networked systems, and therefore it investigates appropriate approaches for dynamic analysis, built on top of the CONNECT monitoring mechanism. Since both the type of properties and the entities that the properties refer to cannot be fixed once and for all, the monitor refers to the refined and composable framework, which is based on the well-established generic metrics. We also need to identify new verification and validation models and techniques able to capture the necessary aspects required by the refined metrics for the four entities belonging to the CONNECT infrastructure. For such challenges we look at the potential offered by dynamic analyses in aiding the understanding, development, and evolution of dynamically CONNECTed systems.

Acknowledgements

This work has been partially supported by the Future and Emerging Technologies (FET) European Project CONNECT (FP7–231167) (www.connect-forever.eu).

8. REFERENCES

[1] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable and Secure Computing*, 1(1):11–33, 2004.

[2] CONNECT Consortium. Deliverable 6.1 – Experiment Scenarios.

[3] CONNECT Consortium. Deliverable 5.1 – Conceptual models for assessment and assurance of dependability, security and privacy in the eternal CONNECTed world, 2010.

[4] G. Di Marzo Serugendo. Robustness and dependability of self-organizing systems - a safety engineering perspective. In R. Guerraoui and F. Petit, editors, *SSS*, LNCS 5873, pages 254–268. Springer, 2009.

[5] D. C. Schmidt. Model-driven engineering. *IEEE Computer*, 39(2), February 2006.