The HIDENETS holistic approach for the analysis of large critical mobile systems

Andrea Bondavalli, Ossama Hamouda, Mohamed Kaâniche, Paolo Lollini, Istvan Majzik, Hans-Peter Schwefel

The HIDENETS Holistic Approach for the Analysis of Large Critical Mobile Systems

A. Bondavalli[1], O. Hamouda[2,3], M. Kaâniche[2,3], P. Lollini[1], I. Majzik[4], H.-P. Schwefel[5,6]

[1]*University of Florence, Dept. of Systems and Computer Science, Florence, Italy*
[2]*CNRS; LAAS; 7 av. Du Colonel Roche, F-31077 Toulouse, France*
[3]*Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France*
[4]*Dept. of Measurement and Information Systems, Budapest Univ. of Technology and Economics, Hungary*
[5]*Forschungszentrum Telekommunikation Wien (FTW), Austria*
[6]*Aalborg University, Aalborg, Denmark*

Abstract— Dealing with large, critical mobile systems and infrastructures where ongoing changes and resilience are paramount leads to very complex and difficult challenges for system evaluation. These challenges call for approaches that are able to integrate several evaluation methods for the quantitative assessment of QoS indicators which have been applied so far only to a limited extent. In this paper we propose the holistic evaluation framework developed during the recently concluded FP6-HIDENETS project. It is based on abstraction and decomposition, and it exploits the interactions among different evaluation techniques including analytical, simulative and experimental measurement approaches, to manage system complexity. The feasibility of the holistic approach for the analysis of a complete end-to-end scenario is first illustrated presenting two examples where mobility simulation is used in combination with stochastic analytical modelling, and then through the development and implementation of an evaluation workflow integrating several tools and model transformation steps.

Index Terms— 3.II.VIII.III Computer Systems Organization---Communication/Networking and Information Technology---Mobile Computing---Mobile communication systems, 9.VI.V.I Computing Methodologies---Simulation, Modelling, and Visualization---Model Development---Modelling methodologies, 10.IX.IV Computer Applications---Mobile Applications---Pervasive computing

1 Introduction

Recent advances in wireless and portable devices technologies have opened new opportunities for innovative services that can be accessed by mobile users in highly dynamic environments, through a combination of ad-hoc and infrastructure based communication networks. Such services cover a large variety of application domains including information and entertainment (voice and video streaming, online gaming, contextual information services, etc.), as well as safety and dependability critical services (hazard warning, safety and traffic management for transportation systems, assisted living support systems and healthcare monitoring, crisis management, etc.) [1]. This fast growing area poses some significant challenges from the dependability point of view that require the development of innovative approaches to support design, validation and assessment activities [2].

In this paper, we focus on the challenges raised by the end-to-end *dependability evaluation of such services and scenarios* and we present an approach developed in the context of the HIDENETS European project [3]. This approach is aimed at providing quantitative dependability and performance Quality of Service (QoS) metrics to support design activities. A fundamental challenge is to *master the complexity* of the systems supporting the delivery of such services, taking into account their heterogeneity (in terms of the technologies used and their dependability and performance characteristics), the large number of components, and the dynamicity of the users.

The proposed approach is designed to *combine different evaluation techniques*, including analytical modelling, simulation and experimental measurements, which can be applied at *different abstraction levels*. Similar principles are widely recognized to be necessary to master the complexity of large networked critical systems and to evaluate their dependability [4].

The main contributions of this paper consist in (1) the presentation of a *holistic approach* (that is based on the above principles) for the dependability evaluation of end-to-end scenarios in critical mobile systems, and (2) the illustration of the feasibility of the holistic approach on case studies from vehicular applications. The essence of the holistic approach is the application of various techniques at different abstraction and decomposition levels to *solve sub-problems*, and exploitation of the *interactions of these techniques* to obtain the solution of the complex problem of end-to-end dependability evaluation. The abstraction levels cover user, application, architecture and communication layers, while the potential interactions, as detailed later, could include cross-validation of assumptions, obtaining partial solutions, and refinement of the problem. The first two case studies show how mobility simulation is used in combination with stochastic analytical modelling to assess some dependability properties of the investigated applications. The third case study presents an example of an evaluation workflow, which illustrates the feasibility of an automated integration of several tools and model transformation steps to support the holistic approach.

The paper is structured as follows: Section 2 provides a

short overview on the HIDENETS project, outlines the main challenges to be addressed to evaluate the dependability of large critical mobile systems, and presents the related work. The proposed holistic approach is described in Section 3, while the three case studies illustrating its feasibility are presented, respectively, in Sections 4, 5 and 6. Finally, Section 7 summarizes the main conclusions of this work.

2 HIDENETS Overview, Challenges and Related Work

This section first provides a short overview on the HIDENETS project [3] to clarify the modelling context. Then, it discusses the specific challenges on the quantitative analysis and presents the related work.

2.1 Quantitative analysis in the HIDENETS project

The HIDENETS project primarily addressed the provisioning of available and resilient distributed applications and mobile services in highly dynamic environments characterized by unreliable communications and components. The concept of *resilience* extends the classical notion of fault tolerance (usually applied to recover system functions in spite of operational faults) to some level of adaptability, so as to be able to cope with system evolution and unanticipated conditions [5].

The investigations in HIDENETS included networking scenarios consisting of ad-hoc/wireless (multi-hop) domains as well as infrastructure network domains. Applications and use-case scenarios from the automotive domain [1], based on car-to-car (C2C) communications with additional infrastructure support, have been used to identify the key challenges, threats, and resilience requirements that are relevant in the context of the project and specifically for the analysis approaches.

A HIDENETS *use-case* is a set consisting of (one or more) applications, the actors and roles involved, and the identification of the affected dependability domains. The identified applications for a use-case are assumed to occur in a certain context where these applications typically appear together and interact with each other. The actors and their roles represent the glue of the use-case and are important for the interaction between the applications. Large part of the research work in HIDENETS was motivated by three main use-cases [1] which impose complementary challenges and functionalities: *Infotainment* with elastic quality requirements, *Platooning*, with strict safety and timeliness requirements, and *Car Accident*, in which a number of applications with high dependability requirements operate in parallel. The latter use-case is the main setting of the evaluation examples in Section 5.

Driven by the challenges and requirements of the use-cases, the HIDENETS project has developed appropriate run-time resilience support via fault-prevention and fault-tolerance mechanisms at the middleware and communication layers. Furthermore, the project adopted appropriate architectural constructs, as well as methodologies to support the design, development, evaluation, and testing of dependable solutions using such mechanisms. An overview of the middleware and communication level services, their design and lessons learned in HIDENETS can be found in Chapter 3 of [6]. In order to evaluate the dependability of end-to-end complex scenarios, adequate *holistic evaluation approaches* have been developed, which are described in the further sections of this paper.

2.2 Challenges in large critical mobile systems

The assessment of the dependability-related attributes of the use-cases and applications in large critical mobile systems is a very challenging topic due to their characteristics which include (see [7]): i) *use of OTS components*, which usually exhibit little information on their development process, on their architecture and on their actual failure behaviour; ii) *dynamicity* in terms of topology, connectivity, and channel conditions; iii) *interdependencies between different system parts*, e.g. resulting from functional or structural interactions between system components; iv) *variety of threats*, including both accidental and malicious faults (attacks and intrusions). In the following we provide some more details on three specific properties that have deeply affected the quantitative assessment activities described in this paper: mobility, heterogeneity and largeness.

Mobility of actors. Mobility contributes to most of the dynamics of the considered C2C environments. This fact makes proper modelling of mobility and its effects on the network domain and application usage a task of crucial importance. Two possible types of mobility patterns can be used and combined to elaborate realistic mobility models: i) *traces* obtained by means of measurements of deployed systems or derived by ad-hoc mobility simulators, and ii) *synthetic models* that correspond to mathematical models abstracting specific characteristics of nodes movements in particular environments.

Heterogeneity of the network domains. The networking scenario includes wireless ad-hoc networks, wireless infrastructure-based networks, and also wired networks. The characteristics of these network domains are quite different. For example, the wireless ad-hoc domain is characterized by high dynamicity, while the fixed network has only low dynamicity (mainly due to network traffic fluctuations and congestion). The heterogeneity could force the modeller to consider *different modelling and solution techniques*, each one specifically tailored to capture the behaviour of a part of the overall system. In this context several challenging issues arise, like the definition of a proper mapping between sub-systems and modelling techniques, as well as the identification of the possible interactions between the different techniques.

Large number of components and scenarios. The set of interacting components involved in a single use-case can be very large, and their number immediately increases if the scale of the system increases as well. The number of components to be considered in the quantitative evaluation process will depend on the level of detail needed to evaluate the quantitative measures under study. Besides the number of components, the complexity of the evaluation also results from the existence of a large number of failure modes and recovery and maintenance scenarios to be taken into account.

2.3 Related work

Several approaches were already proposed to master the complexity of the evaluation of large systems.

Model decomposition partitions a system-level model into a set of simpler and more tractable sub-models, and the measures obtained from the solution of the sub-models are aggregated to those of the overall model. Most decomposition and aggregation methods use a hierarchical (top-down) decomposition to avoid the generation of large models. These approaches allow the evaluation of quantitative measures characterizing the dependability of the target systems at different abstraction levels. Various examples of modelling approaches based on this idea are proposed in the literature (see e.g. [8]).

Analytical state-space stochastic models are commonly used for dependability modelling of computing systems. They are able to capture various functional and stochastic dependencies among components, and allow evaluation of various measures related to dependability and performance based on the same model when a reward structure is associated to them. To master complexity, a modelling methodology is needed so that only the relevant system aspects need to be detailed, allowing numerical results to be effectively computable. A survey on the approaches dealing with model complexity can be found in [4].

With respect to coupling of different modelling techniques, existing attempts in the literature frequently use simulation models (or experimental setups) of low-level system behaviour to obtain parameters to be used by higher-level analytical models (see e.g. [9]).

Other works concern the construction of analysis models on the basis of measurements performed in a running prototype or in a full deployment. In [10], for example, software performance models of distributed applications are extracted from traces recorded during execution. A similar approach is recording error propagation traces induced by fault injection experiments [11] to support the construction of error propagation models or derive high-level behavioural models. More general approaches have been developed in [12] focusing on the interactions between modelling and experimentation for dependability benchmarking.

Regarding the specific lower-level sub-problem of connectivity analysis in ad-hoc networks, a large part of the existing research work focuses on static snapshots of the node placement: [13] analyses different connectivity metrics under the assumption that the node placement can be described by a spatial renewal process. This setting is generalized in [14] to cases of spatial correlation. However, for the use-cases considered in this paper, the dynamics of the connectivity changes are a major factor. This problem has been seldom investigated in the context of vehicular ad-hoc networks, though some recent work has been done in this direction in the context of encounter-based protocols [15]. Note that such studies require the definition of mobility models that are representative of realistic traffic scenarios. A survey of recent initiatives aiming at this objective in the context of vehicular applications is presented in [16].

3 The Holistic Framework

The challenges discussed in Section 2.2, together with the necessity of continuous assessment activities during all the design and development stages of critical mobile systems, call for a composite verification and validation framework where the synergies and complementarities among several evaluation methods can be fruitfully exploited. In the quantitative assessment of such systems, a single evaluation technique (including analytical modelling, simulation and experimental measurement) is not capable of tackling the whole problem, i.e., the dependability evaluation of end-to-end scenarios. To master complexity, the application of the holistic approach allows defining a "common strategy" using different evaluation techniques applied to the different components and subsystems, thus exploiting their potential interactions. The idea underlying the holistic approach follows a "divide and conquer" philosophy: the original problem is decomposed into simpler sub-problems that can be solved using appropriate evaluation techniques. Then the solution of the original problem is obtained from the partial solutions of the sub-problems, exploiting their interactions. Some of the possible interactions among different evaluation techniques are the following:

- *Cross validation*. A partial solution validates some assumptions introduced to solve another sub-problem, or validates another partial solution (e.g., a simulation model can be used to verify that the duration of an event in an analytical model is exponentially distributed).
- *Solution feedback*. A partial solution (or a part of it) obtained by applying a solution technique to a sub-problem is used as input to solve another sub-problem possibly using a different technique (e.g., a critical parameter in an analytical model is obtained using experimental evaluation).
- *Problem refinement*. A partial solution gives some additional knowledge that leads to a problem refinement (e.g., the architecture of a component changes since it is recognised to be a system bottleneck).

It is clear that the system decomposition is not unique, as we can identify different system decompositions corresponding to different levels of abstraction. The higher the level of detail required to capture the system behaviour, the higher is the complexity of the system to be modelled and solved. Therefore the choice of a particular system decomposition is of primary importance, and it is always a trade-off between faithfulness of representation of the real system behaviour (with respect to the measures of interest) and capability to solve the corresponding models. In the following, we depict a type of system decomposition, the abstraction-based system decomposition, which statically focuses on the various levels of abstractions that can be used to represent a system.

The overall system can be analyzed at different levels of abstraction: each level captures a specific aspect of the overall system behaviour and it "communicates" with the other levels through some well-specified interfaces. Such interfaces mainly define the input they require from other abstraction levels, as well as their output. The proposed decomposition is also useful to understand and quantify how certain faults occurring at the lower levels of the

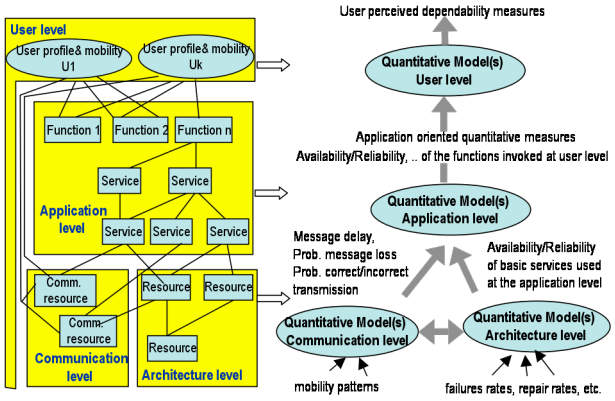

Fig. 1. Levels of abstractions.

hierarchy can propagate and affect the higher levels, leading to erroneous states or failures when such errors reach the user. In particular, we have identified the following abstraction levels, depicted in Figure 1 (left side).

User level. This level provides high-level QoS and dependability attributes as perceived by the users. It describes the users' profiles, that is, how the users interact with the application and how their requests are mapped to the different components of the architecture. Accordingly, the QoS and resilience attributes perceived by the users depend on the QoS and resilience attributes of the corresponding components. A user level is needed to account for different classes of users having different behaviours and different requirements. Mobility scenarios and application utilization profiles are just some examples of users' characteristics that can differentiate a user's class from another. Note that mobility scenarios directly influence network dynamics, this way the underlying communication resources.

- The expected inputs are the outputs produced by the application level.
- The expected outputs are high-level QoS attributes related to the user's perspective. Examples of user-oriented measures include availability, safety or reliability reflecting the considered use-cases and operational profiles.

Application level. This level describes the system behaviour from the logical and functional point of view. The applications differ in their technical properties, their mechanisms, their interfaces, and they can impose different communication and middleware level requirements. The user-interfaces consist of a set of functions, and each function corresponds to a set of (middleware) services offered by the architecture for its implementation. Each function may depend on several services and the services may depend on each other.

- The expected inputs are the outputs produced by the architecture and communication levels.
- The expected outputs are QoS and dependability related measures associated to each function invoked at this level, like availability, reliability, etc. Some of these measures could be provided as input to the user level.

Architecture level. This is the part of the system capturing the behaviour of the main hardware and software components (resources) that can affect the application-level measures, including the error detection and recovery mechanisms implemented in the system to support dependability and resilience. It describes how the functions and services of the application level are implemented on these resources. This layer also includes the middleware that abstracts some details of the underlying layers for the application running on top.

- The expected inputs are some low level parameters concerning hardware, software or basic services, such as failure rate, error latency, repair rate, error propagation probability.
- The expected outputs are some medium-level dependability-related attributes, like availability and reliability of some services used at the application-level.

Communication level. It captures the communication aspects of the system that can affect the application level. It addresses the link layer (possibly considering several types of networks like WLANs, UMTS and GPRS), the network layer (IP-based) and the transport layer (considering several types of protocols like TCP and UDP).

- The main expected inputs are mobility scenarios (from the user level), the traffic patterns, and a set of assumptions introduced to hide low-level system details that are not the target of the analysis.
- The expected outputs are communication level measures like: message delay, probability of lost message, probability that a message is incorrectly emitted or is omitted. Such measures could be mean values or complete distributions, and could be used at the application-level or at the architecture level.

Indeed, the communication level can be seen as a special case of the architecture level focusing on some communication related aspects that might affect the QoS and resilience characteristics perceived at the application and the user levels. Also, it is noteworthy that sometimes we might need to analyze and assess some characteristics of the application level as a function of communication related aspects without explicitly modelling in detail the architecture level. Similarly, depending on the level of detail considered in the description of the studied system, it could be sometimes more efficient to model different levels (e.g. the user and the application levels) as a single abstraction level.

To evaluate the QoS and dependability measures associated to the different abstraction levels discussed above, one or several dependability sub-models can be associated to each level (see right side of Figure 1). Each sub-model can be processed using inputs evaluated from lower level sub-models, besides other inputs provided e.g. by measurement. It can also be processed independently of lower level sub-models by making assumptions about the behaviour and the parameters characterizing the dependability of lower level components. The selec-

tion of the appropriate modelling and evaluation techniques for each level (Markov Chains, Petri Nets, queuing networks, simulation, etc.) also depends on these assumptions and on the quantitative measures to be assessed.

Some examples illustrating the combination of different techniques within the holistic framework are presented in the following sections. The first addresses the estimation based on simulations of some connectivity parameters used in the analytical dependability models associated with the Distributed Black Box application (Section 4). The second example integrates the output traces generated by an ad-hoc mobility simulator into analytical models capturing a subset of the Car Accident use-case scenario (Section 5). Finally, we will also provide a case study that demonstrates the integration of several tools and model transformation steps to support the holistic approach (Section 6). The models at different levels are aggregated and higher level models are generated in an automated way, forming an *evaluation workflow* that defines how and where such integration can be realized.

4 The Distributed Black Box Application

Similarly to avionics black-boxes, the Distributed Black-Box (DBB) application investigated in the context of the HIDENETS project provides a virtual mechanism to record periodically historical data about the state of participating vehicles and their environment, which can be replayed in the event of an accident [17]. To protect the data against accidental and malicious threats, this data is temporarily replicated on participating vehicles encountered in the ad-hoc domain. Permanent backups are created when the vehicles (the data owner or the participating vehicles) access the fixed infrastructure.

Compared to the scenario where the data is permanently stored only when the vehicle collecting the data gets access to the fixed infrastructure, the efficiency of this service from a dependability viewpoint depends on a number of environmental factors, like the density and mobility characteristics of participating vehicles, the density of Internet access points for all vehicles, the occurrence rate of accidental failures and potentially malicious contributor behaviour affecting the vehicles.

This section illustrates how a combined analytical and simulation modelling approach can help to gain better insights into how these issues affect the dependability of the DBB application. Firstly, simulation is used to characterize the distribution of some connectivity parameters in vehicular communication scenarios. It is shown that under certain assumptions the car-to-car and car-to-infrastructure encounter processes can be described by a Poisson process. These parameters are then incorporated into a Generalized Stochastic Petri Net (GSPN) model to assess the impact of permanent failures on the availability of the data. Referring to the abstraction-based decomposition approach presented in Section 3, three abstraction levels are considered for the modelling of the DBB application: 1) the user level defining the mobility patterns and scenarios to be considered in the analysis, 2) a communication level model which produces as an output the distribution of the car-to-car and car-to-infrastructure encounter processes based, and 3) an application level model that uses these connectivity distributions as an input and models the impact of failures on the availability of the data.

This section is organized as follows: Section 4.1 presents background information about the data replication strategies investigated for the DBB application. Section 4.2 presents the simulation-based connectivity analyses considering a two lane freeway mobility scenarios. Section 4.3 presents the GSPN model and examples illustrating the impact of several environmental parameters on data availability.

4.1 Cooperative backup service

The cooperative data-backup service takes advantage of the resources available in a mobile node's neighbourhood to temporarily replicate critical data in the ad-hoc domain; such replicated data can then be permanently backed-up as soon as the participating nodes have an access to the fixed infrastructure. This scenario assumes that the encountered vehicles accept to contribute to the cooperative backup service and implement the middleware needed to run the service. Such contribution can be imposed by law for safety reasons or motivated by lower car insurance rates.

The cooperative backup service also takes care of the data recovery phase by providing algorithms allowing the restoration of the data at the fixed infrastructure. Various data replication strategies can be considered for implementing a cooperative data backup service. We have considered erasure codes that are well suited to ensure data availability and confidentiality in the presence of permanent failures affecting the data. Such failures could be caused by accidental hardware or software faults, or by malicious actions (see [18] for more details).

Given a data item to be encoded with an erasure code (n,k), the algorithm produces $n \geq k$ *fragments*; m fragments are necessary and sufficient to recover the original data item, where $k \leq m \leq n$. When $m = k$, the erasure code algorithm is said to be *optimal* [19]. Simple replication of the data corresponds to the case $k=1$. When all fragments are stored on different vehicles, an optimal erasure code allows n-k failures (or erasures) to be tolerated (besides that of the primary replica). Additionally, when all fragments are distributed to different neighbouring vehicles belonging to different non-colluding users, erasure codes can be regarded as a means for improving data confidentiality: to access the data, an attacker must have access to k fragments stored on different vehicles instead of just one when simple replication is used. Later in Section 4.3 we show the impact of n and k on the dependability of the data that can be offered by a cooperative backup service. The metric considered for assessing data dependability is the probability of data loss in the presence of failures affecting the vehicles involved in a cooperative backup scenario.

4.2 Estimation of connectivity parameters

This section addresses the estimation based on simulations of the properties of some connectivity parameters

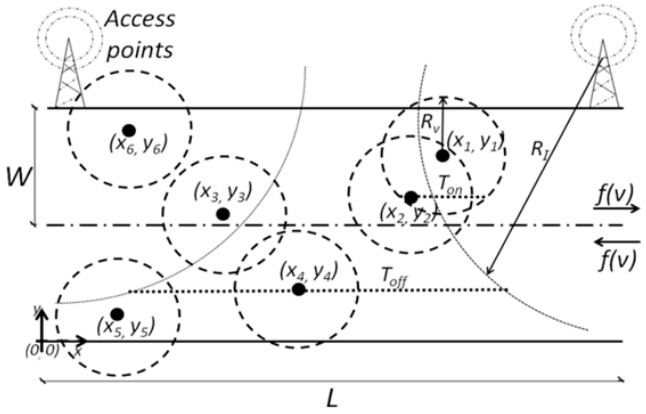

Fig. 2. A two lane freeway mobility scenario.

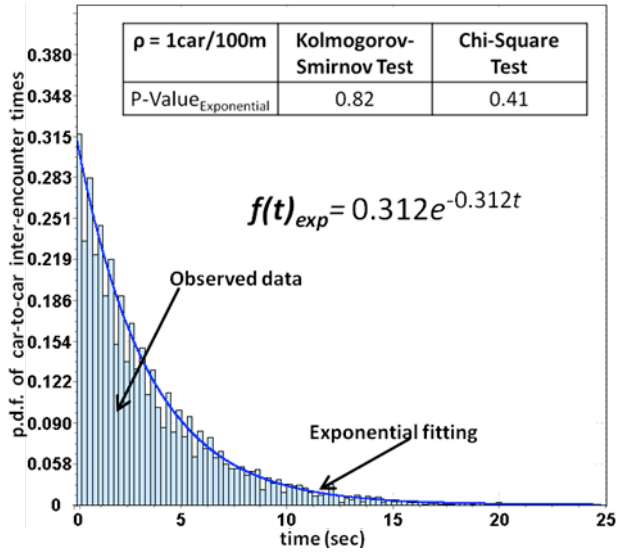

Fig. 3. Empiric probability density function of the time between single-hop encounters for, car density ρ = 1car/100m: simulation results and comparison to an exponential distribution.

characterizing vehicular communication scenarios. These parameters are needed to evaluate the dependability of the DBB application using analytical models. Such parameters are the rate at which vehicles meet, the duration that such vehicles stay in connectivity range, and the rate at which a vehicle meets an access point of the fixed infrastructure.

Figure 2 shows an abstraction of a vehicular freeway scenario used in our context to estimate such connectivity parameters. We consider a long straight piece of freeway (of width W) with movements in two directions. The considered piece of freeway has a finite length $L \gg W$. In order to avoid edge effects, we assume in simulations that cars that leave on one side enter at the corresponding point on the other side, *i.e.*, the long piece of road can be seen wrapped around a cylinder. It is assumed that vehicles have constant speed v_i, however these velocities v_i could be different among the vehicles and they are assumed to be identically and independently distributed according to a probability density function $f(v)$.

In order to focus on understanding the impact of the geographic mobility model, we adopt the approximation that two nodes can communicate on a direct link when their geographic Euclidean distance is less than a communication radius R_v, where R_v is a constant for all nodes regardless of speeds. Hence, we assume a homogeneous communication technology. It is an advantage of the employed simulation approach that more complex link-layer connectivity models including Doppler shifts and multipath propagation effects in changing physical environments can easily be included. To characterize connectivity dynamics, we consider a reference vehicle located at position (x_1, y_1), and we analyze two main processes:

1) *Car-to-Car encounter process* that models the time instances at which other cars enter (single-hop or s-hop) connectivity to the reference car. For s=1, these instances correspond to new cars coming into radio range R_v of the reference car. The mean time between such encounters is represented by α^{-1} and α is called the car-to-car encounter process rate, assumed to be constant.
2) *Car-to-Infrastructure encounter process* that models the time instances at which a vehicle comes into radio range R_I of an access point of the fixed infrastructure. The mean time between such encounters is represented by β^{-1} and β is called the car-to-infrastructure encounter process rate, assumed to be constant.

Other processes can also be investigated, e.g., to characterize the duration of connectivity periods. Detailed results are presented in [20]. In the following, we present two main results that are needed for the analytical dependability model presented in Section 4.3.

The first result presented in Figure 3 concerns the distribution of the times between single hop car-to-car encounters. The simulation is performed using the following set of parameters:

- Each vehicle has a constant speed selected at the beginning of the simulation uniformly distributed between $v_{min} = 80km/h$ and $v_{max} = 130km/h$.
- Each vehicle is assigned x coordinate between (0, L) and y coordinate between (0, $2W$) according to a uniform distribution ($L = 5000m$ and $W = 15m$).
- The reference vehicle at the beginning of the simulation is assigned initial coordinates $(x_1, y_1) = (2500m, 22.5m)$ and speed $v_1 = 108km/h$.
- The car density is $\rho = 1car/100m$.

Note that for this set of parameters, the communication range of the reference vehicle covers the full width of the freeway in both driving directions.

The movement of the cars is simulated considering fixed time steps of granularity $0.1sec$. Time steps at which k-hop connectivity relations to the reference car are established newly or vanishing are recorded. The results from the simulation are used to investigate the distribution of the car-to car inter-encounter times.

Figure 3 plots the empiric probability distribution function for the inter-encounter time in the single-hop case obtained from samples from 300 simulation runs, each entailing $5hrs$ simulated time (approximately 600 encounter samples in each run). Statistics for the times between encounters observed from the simulation show a

mean = 3.34*sec* corresponding to a rate estimate $\hat{\alpha}$ = 0.29/*sec* and a variance of inter-encounter times = 10.38*sec*^{2}. The encounter rate value that results from the least-square fit to an exponential distribution (as also taking the mean estimate which is previously stated is a way of fitting, namely the maximum likelihood fit) is α=0.312/*sec*, so rather close to the simulation estimate. This is confirmed by the P-values associated to the Kolmogorov-Smirnov and χ^2 statistical tests shown in Figure 3.

It is noteworthy that an analytical proof provided in [20] shows that the encounter process is a Poisson process when considering an infinite freeway lane with an initial placement of cars according to a spatial Poisson process. Additional results illustrating the impact of the radio range R_v as well as investigations to what extent a Poisson process is still a good approximation in more complex settings are also presented.

The second result concerns the distribution of the car-to-infrastructure inter-encounter times. Besides the simulation parameters used for Figure 3, the density of access points is assumed to be equal to *1/km* with a radio range R_I = *250 m*. The plot of the empiric inter-encounter distributions looks qualitatively similar to Figure 3 and hence is omitted here. Statistics from the simulation for the car-to-infrastructure inter-encounter times show a *mean* = 81.1 *sec* corresponding to a rate estimate $\hat{\beta}$= 0.0123/*sec*. The encounter rate that results from the fitting to an exponential distribution is β=0.011/*sec*. Also, the good quality of fit of the exponential distribution to the data is confirmed by the Kolmogorov-Smirnov and χ^2 tests. It is noteworthy that a similar conclusion is obtained when considering a lower density of access points (e.g., 1/2.5km or 1/5km).

4.3 Dependability modelling

This section presents an analytical model based on Generalized Stochastic Petri Nets (GSPNs) aimed at assessing the combined impact of failures and mobility characteristics on the dependability of the data in the context of a cooperative backup service.

GSPNs are commonly used to perform dependability evaluation studies and sensitivity analyses aimed at identifying parameters having the most significant impact on the measures. The corresponding models are based on the assumption that all the underlying stochastic processes are described by exponential distributions.

In the previous section, we have shown that the exponential distribution is an acceptable assumption for describing connectivity parameters when considering the investigated freeway mobility scenario. As regards the distribution characterizing inter-failure times, the exponential distribution is a common assumption in dependability studies.

Figure 4 presents a generic GSPN model of the cooperative backup service using an *(n,k)* erasure coding algorithm. This model is presented in detail in [18]. Here we only summarize the main characteristics needed to illustrate the combination of results obtained from the simulation and the analytical modelling. The model focuses on the mobile *ad-hoc* part of the cooperative backup service, assuming that the infrastructure-side functionalities are reliable. We describe the behaviour of a data item in the presence of failures from the time it is generated at the source vehicle (called owner) until it is saved at the infrastructure side or it is completely lost. A data item is considered "safe" (Place DS is marked) whenever either its owner or a participating vehicle storing it (called contributor) is able to access the Internet. Thus, with (*n,k*) erasure coding, a data item is definitely lost (Place DL is marked) *if and only if* its owner vehicle fails *and* less than *k* contributors hold or have held a fragment of the data item. This condition is specified in the predicate *L* attached to the transitions of the Petri net. The failures of the owner and the contributors are represented by transitions λ_0 and $m(MF)\lambda$, respectively. *m(MF)* corresponds to the number of tokens in place MF indicating the number of contributors holding a fragment of the data. Car-to-car encounters are represented by transition α, and car-to-infrastructure encounters are represented by transition β_0 for the owner and $m(MF)\beta$ for the contributors. The associated rates are derived from the simulation experiments presented in Section 4.2.

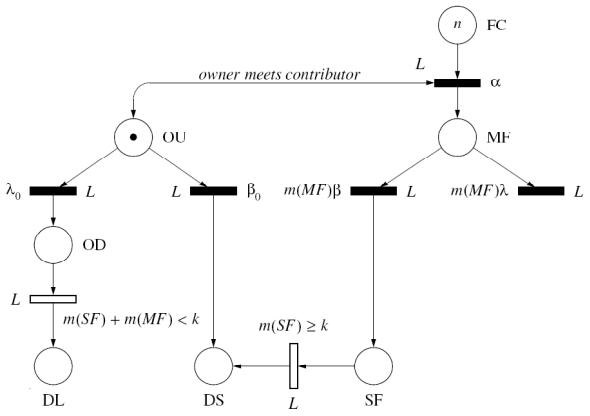

Fig. 4. GSPN model describing the cooperative backup of a data item in the presence of failures.

The dependability of the data backup service can be assessed *via* the evaluation of the probability of data loss, i.e. the asymptotic probability, noted *PL*, of reaching the *DL* place. Let us denote by PL_{ref} the probability of data loss corresponding to the non-cooperative backup scenario (i.e., the data is not replicated in the ad-hoc domain and is backed up only when the owner accesses the infrastructure). We can measure the dependability improvement provided by a cooperative backup service compared to a non–cooperative backup scenario by evaluating the data loss probability reduction factor $LRF = PL_{ref}/PL$.

4.4 Example of results

Figure 5 shows an example result illustrating the data dependability improvement yielded by the cooperative backup service by plotting the evolution of data loss factor *LRF* as a function of the failure rate λ, considering different values of α/β and *n*. Here we consider the case of simple replication (*k=1*) and we assume that owners and contributors behave similarly ($\beta_0=\beta$, $\lambda_0=\lambda$).

The results in Figure 5 are obtained with the parameters α and β estimated from the simulation experiments. Three different cases are distinguished corresponding to

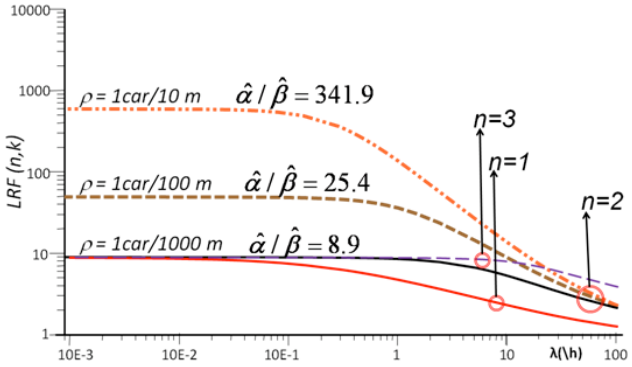

Fig. 5. Data loss reduction factor as a function of λ, considering different parameter values for α/β (*k* =1 in this case).

three mobility scenarios with car density $\rho = 1car/10m$, $\rho = 1car/100m$, and $\rho = 1car/1000m$, respectively. The connectivity ratios α/β estimated for these three cases are of the order of magnitude 341.1, 25.4 and 8.9 respectively. The data dependability improvement compared to the non cooperative backup scenario depends on the value of the failure rate λ. It can be seen that the maximum gain is of the order of magnitude of the ratio α/β, which depends on the considered mobility scenario.

5 The Car Accident Use-case

The car accident use-case scenario evolves around a scene with an accident on a road, involving cars. The analyzed network scenario is composed by a set of overlapping UMTS cells covering a highway, and a set of mobile network devices (embedded or inside cars and emergency vehicles) moving in the highway and requiring different UMTS classes of service. The concrete UMTS scenario under analysis is depicted in Figure 6.

Four base stations are considered: A, B, C and D. The users (cars) are moving in two different road lanes: some in the left to right lane (from A to D) and the remainder in the right to left one (from D to A). We assume that the accident occurs in the C zone, in the left to right lane, forcing other users approaching that area to stop until the ambulance arrives, the crash site is cleaned and the normal traffic flow restored. The emergency vehicle heads

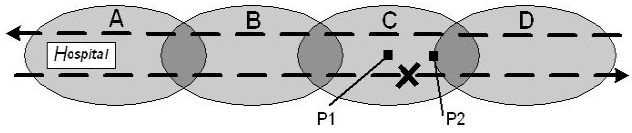

Fig. 6. The analysed scenario.

back to the hospital towards the A zone, where we suppose the hospital is located. Concerning the available UMTS services, we suppose that a generic user can use three different services (Telephony, Web Browsing and File Transfer), while the ambulance uses the "access to medical expertise" application that consists of two simultaneously running services (Emergency Streaming to transmit the ECG traces, and Emergency Videoconference to fully interact with the hospital), having higher requirements in term of signal to interference ratio as compared to the nonemergency services. The services mainly differ by the activity factor, the uplink and downlink throughput and the required signal-to-interference ratio.

The metrics of interest concern the QoS levels both from the users' perspective and from a mobile operator's point of view. Typical user-oriented QoS indicators are the probability that a service request is successfully completed (P_{succ}), blocked (P_{block}) or dropped (P_{drop}). Typical mobile operator-oriented indicators are the load factor, both in uplink (η_{ul}) and downlink (η_{dl}), and the number of allocated traffic channels, which corresponds to the average number of served users.

The Car Accident use-case scenario has been analyzed in [21] through a modelling approach based on the Stochastic Activity Network (SAN [22]) formalism. The focus was on three UMTS characteristics having important effects on the QoS: the random-access procedure, the admission control strategy and the soft handover mechanism. These characteristics mainly influence the so called "connection level" QoS, which are the quality indicators related to the connectivity properties of the network, like the call blocking or dropping probability. Exploiting the modularity of the modelling framework, in [23] the same authors defined the approach for integrating the output produced by an ad-hoc mobility simulator into the modelling process itself, which allows capturing more complex and detailed mobility dynamics that can heavily affect the analyzed QoS indicators. Such interaction constitutes a concrete example of an application of a holistic evaluation approach, where the synergies and the characteristics of different evaluation techniques (here mobility simulators and SAN models) are exploited to capture system characteristics at a more detailed level, thus enabling a more refined QoS analysis that could be hardly obtained using a single technique.

Building on these previous works, in the remaining of this section we will first focus on the SAN - mobility simulator interaction (Section 5.1), also showing how it allows a refinement of the UMTS network model (Section 5.2). Then we will present some results showing the impact of the mobility and network model refinement on the selected QoS indicators (Section 5.3).

5.1 Refinement of the mobility aspects

The UserMobility SAN model presented in [21] represents the user movement across the UMTS network scenario, and it belongs to the 'User' abstraction level of Figure 1 (it defines the mobility characteristics of the users). The mobility pattern of every single user included in the scenario is modelled by an instance of the UserMobility model. In accordance with the modelling assumptions, the mobility scenario is represented through different "zones", characterized by a given active set of available base stations, and the user moves through these zones in a uniformly distributed time that depends on the cell size and on the average speed of the user.

As shown in Figure 7, each UserMobility model can be logically split in two distinct parts: the "Mobility Pattern" part and the "Translation" part. The "Mobility Pattern" part represents the stochastic mobility scenario imple-

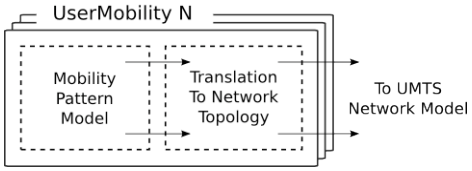

Fig. 7. The two logical components of the UserMobility atomic model.

menting the user mobility rules and updating the user position when required. There is a place for each zone and uniformly distributed activities to model the movement between them. The "Translation" part performs a mapping between the user position in the scenario and the network topology, i.e., it identifies the available base stations based on the current user position (the zone where the user is located).

A more refined representation of the user mobility aspects can be achieved by allowing the SAN model to read and use the detailed mobility traces generated by a mobility simulator or collected from real-life experiments. Thanks to the modularity of the model and the clear separation between the two roles of the UserMobiliy model, the goal is achieved with few modifications to the original model (see Figure 8).

As first step, the "Mobility Pattern" part of the UserMobility models is replaced by some interface places, which hold the current user position and serve as input to the "Translation" part. These interface places will get their values from the mobility trace. An additional atomic model, the TraceParser model, is then introduced to read the trace file and fill the proper values in the interface places. In the current implementation the mobility traces have been generated using VanetMobiSim[1], a Java-based

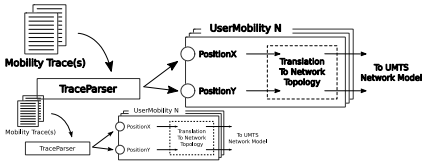

Fig. 8. Integration between the SAN model and a mobility simulator, using traces.

mobility simulator which reads the scenario definition from an XML file and generates a trace file with the following format: NodeID, Time, XPosition, YPosition.

The metrics of interest have been computed through the interaction between the mobility simulator and the discrete event simulator provided by Möbius [24] tool, as depicted in Figure 9.

Before the execution of each Möbius simulation batch (trajectory i, with $i=1,\ldots,n$, where n is the number of batches), the mobility trace file is updated with a new one, stochastically generated by the mobility simulator. The stochastic mobility scenario is then represented by the set of generated mobility traces. Given a (transient) measure of interest M (e.g., the cell load factor at a given instant of time), for each trajectory i an observation O_i is

[1] http://vanet.eurecom.fr/

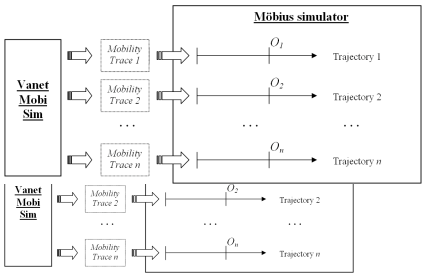

Fig. 9. VanetMobiSim - Möbius interactions.

computed, which corresponds to a sample point of M; then, for example, its mean can be computed as $\sum_{i=1}^{n} O_i / n$, and confidence intervals can be generated.

The mobility trace needed for a single simulation batch is generated and then read by the Möbius simulator itself through a fully automatic process, without additional libraries or dependencies. The whole process is handled by the additional TraceParser atomic model, which invokes the mobility simulator at the beginning of the simulation and then reads the generated trace at periodic intervals (matching the values of the Time field), until the Möbius simulation ends. The mobility simulator is executed using a blocking system call, therefore the Möbius simulation is suspended until the mobility trace has been generated. Following this approach any mobility simulator or even experimental measurement tools could be used, provided that they are capable to generate mobility traces as output.

5.2 Refinement of the UMTS network model

At this point we can use this additional information, i.e., the exact position of the users in the network topology, to refine the UMTS network model through a more refined load factor estimation. With respect to Figure 1, the UMTS network model mainly represents the communication level aspects, although it also abstracts some basic architecture and application level elements. The considered admission control algorithm is based on the load factor of the UMTS cell: a new call is accepted if the load factor level reached after adding the call does not exceed a pre-specified threshold, both in uplink and in downlink. That is:

$$\eta_{ul} + \delta_{ul} \leq \eta_{ul\_threshold} \quad (1)$$
$$\eta_{dl} + \delta_{dl} \leq \eta_{dl\_threshold} \quad (2)$$

where η_{ul}, δ_{ul} and $\eta_{ul\_threshold}$ (or η_{dl}, δ_{dl} and $\eta_{dl\_threshold}$) are, respectively, the cell load factor before the admission of the new call, the load factor increment due to the admission of the new call and the pre-specified threshold level in uplink (or downlink). According to well-known UMTS equations (e.g., see [25]), the load factor increment in downlink and uplink for a given user and service can be computed (within the UMTS network model) as:

$$\delta_{dl} = \frac{(E_b/N_0)\cdot R\cdot \nu}{W}\cdot(1-\alpha+i) \qquad (3)$$

$$\delta_{ul} = \frac{1}{1+\frac{W}{(E_b/N_0)\cdot R\cdot \nu}}\cdot(1+i) \qquad (4)$$

where E_b/N_0 is the required service quality, R is the service data rate, υ the service activity factor, α the average orthogonality factor, W is the chip rate and i is the other-to-own interference ratio at current user position.

While most of the parameters in Equations (3) and (4) are directly obtained from the service class or other network variables, the i factor depends on the power of received signals, which in turn depends on the user position in the scenario. To perform the admission control procedure, the load factor increments have to be computed for each user and each network service.

In [21] it has been assumed that users which are using the same service in the same zone generate the same (fixed) amount of load (and interference) on the involved base station(s). In other words, the i parameter (i.e., the other-to-own interference ratio) has been set to an average value *not depending on the current user position*. In a zone covered by two or more base stations a user can take advantage of soft handover and connect to two or more base stations. When this happens, the load generated on each base station is lower (but still fixed) than the load that would be generated having a single connection.

Since the refined mobility model provides the exact user positions, we can now refine the load factor estimation within the UMTS network model. The concept of *path loss* describes the signal propagation in the modelled environment. Among other factors, it is a function of the distance between nodes and its calculation varies based on the selected propagation model. For simplicity we will consider the free-space path loss (see [25]), which is given by:

$$L_{\mathrm{db}} = 32.44 + 20\log f + 20\log d \qquad (5)$$

where f is the operating frequency in Mhz and d is the distance in Km. Assuming that the total transmitted power of all base stations in the area is the same, the i factor for a given user can be computed as a function of the path losses between the user and the base stations [26]. The i factor for a mobile m served by cell j can thus be computed as: $i(m)=\sum_{k\neq j}(L_{km}/L_{jm})$, where L_{km} is the path loss between base station k and mobile m.

5.3 Numerical evaluations

In this section we evaluate and compare the QoS indicators obtained solving the basic and the refined models.

We consider the car accident event occurring at time t=4001 sec., which is cleared 1000 seconds later. As mentioned before, to compute the i factor in the trace-enhanced version the free-space path loss formula is used (Equation (5)).

In Figure 10 we compare the uplink load factor of base station C obtained using the basic and the refined models (the downlink factor has a similar trend). The two vertical lines represent the instants of time when the car accident occurs and it is cleared, respectively, while the horizontal one represents the maximum allowed cell load factor in uplink ($\eta_{ul\_threshold}$). Considering the refined models, we present two analyses corresponding to two different car accident coordinates: the first one with the accident occurring in the center of base station C (i.e., where the antenna is located), and the second one with the accident occurring at the border of the overlapping area between cells C and D (points P1 and P2 of Figure 6, respectively). We consider a total of 50 cars moving in the scenario, with an average speed of 90 Km/h when not involved in the traffic jam caused by the accident. Analyzing the results we first note that the differences between the "basic" and the "refined" plots are really significant. The refined estimations show that the same number of users camped in the cell is actually producing a lower load factor. This means that, with the adopted setting, the non-refined evaluation process overestimates the load factor of the cell. In addition, considering the "refined" plots, we note that an accident occurring near the border of the cell (the "BORDER" plot) initially has a higher impact on the load factor due to the higher interference produced by the users, and then it becomes lower once the car accident is cleared, because the users are near the border and then immediately enter in the overlapping area with base station D.

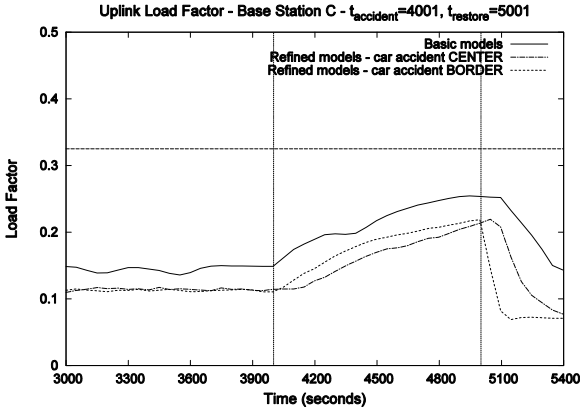

Fig. 10. Load factor (uplink) of base station C.

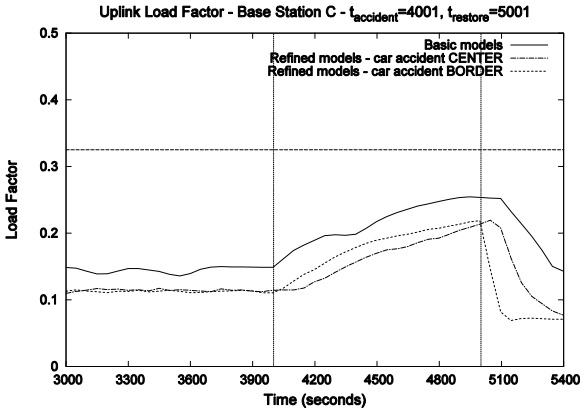

Fig. 11. Probability that a "Browsing" service request is blocked.

In Figure 11 we compare the probability that a Web Browsing service request is blocked as time elapses, for the two different car accident coordinates. In accordance with the previous discussion, the values obtained through the solution of the "refined" models are lower, and for

t=5000 seconds (just before the accident is cleared) the blocking probability for the "BORDER" scenario is about two times the value of the "CENTER" one. The results for the other services have similar trends.

6 Evaluation of the Success of User Activities Using an Automated Evaluation Workflow

In this section we present a study that computes a user-level dependability attribute, namely the probability of the successful execution of user activities in a dynamic scenario [27]. In this scenario the user (driver of a car moving along a route) relies on available applications, and tries to execute several activities in an environment that is characterised by changes: the activities include collaboration with other users, the users may move, and the mobility and network traffic influence the availability and quality of the services. In the following we refer to *scenario* as a concept that involves all user-related and environment-related changes.

The evaluation approach is general in the sense that it does not apply only to a specific use-case (i.e., utilization of specific functions and services); instead, the inputs of the evaluation include models that could specify various user activities, describe the structure of the used applications (functions), the dependability parameters of specific services or resources included in these applications, and the road traffic and mobility pattern.

Accordingly, the evaluation approach needs the integration of several tools that are responsible for the construction, refinement and solution of partial models, and aggregation of these into system-level models. The integration of the tools, i.e., the mapping of the outputs of tools to the inputs of successive tools, can be automated by model transformations that implement either syntactic mappings or property-preserving translations between different formalisms. This way an automated evaluation workflow can be implemented.

The inputs and output of the evaluation workflow as well as the internal processing steps are summarized in Figure 12. Parallelograms represent models, and multi-layered parallelograms stand for multiple models. The rectangles illustrate internal model processing and model solution steps. The three sets of input models that represent different views of the scenario are as follows:

- Each *user workflow* specifies the user activities in terms of application utilization. The mobility aspects are not addressed in the user workflow as they are included in the topology model. There is a user workflow for each participant of the scenario. The information required for the evaluation can be extracted either from UML activity models extended with time information (that specify the ordering and relation of user activities using a flowchart-like notation) or from domain-specific workflow models.
- Each *topology model* represents the information on the evolving ad-hoc topology of potential network connections. In case of multiple networking technologies there is a separate topology model for each technology that can be used for communication. The topology model can be constructed utilizing existing mobility trace generator and network topology generator toolchains [28]. As a specific example, VanetMobiSim is supported as mobility trace generator and the topology model is constructed from its output traces.
- The *application-service dependency models* define how the used applications depend on the services, hardware or software components of nodes. These dependencies are described by UML object diagrams. Objects represent the software and hardware components at the underlying architecture level, and links represent the relations among them. The types of components, the redundancy schemes and the component level dependability attributes are modelled using UML stereotypes and tagged values according to the conventions developed in [29].

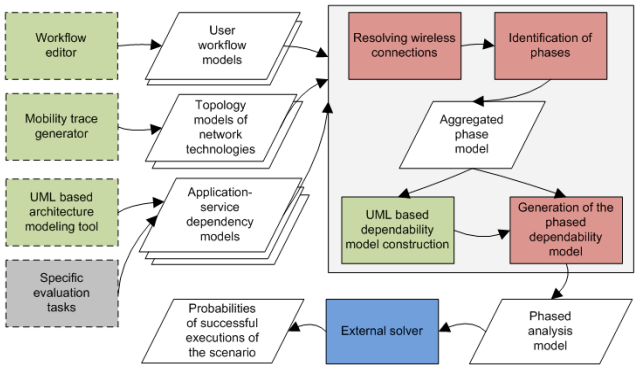

Fig. 12. The evaluation workflow.

The output of the internal dependability model construction steps, a phased analysis model, is solved by an external solver (Möbius or DEEM [30]) which carries out the transient solution resulting in the success probability of the user activity sequence along the scenario. This result can be used to characterize the user workflow in a best case or worst case situation (e.g., whether it is possible to execute the activities in case of extreme network conditions), or to compare different environment options. Another aspect is the comparison of different execution strategies at the user level (e.g., whether it is reasonable to rely on given functions). These results can also be used to synthesize design patterns for given application scenarios, helping the software engineers utilizing the architectural solutions to build more dependable applications. In the following we describe how the concepts of the holistic framework (Section 3) appear in this evaluation workflow. Note that the focus of the description in this section is the integration of models and tools, and not the presentation of concrete evaluation results.

6.1 Hierarchical modelling

The evaluation workflow follows the approach of multi-level modelling addressing phases. Hierarchical multi-level modelling is applied by considering user, application, architecture and communication levels (see Section 3), while the multi-phase approach is followed to separate *phases*, i.e., time periods in which the users' activities and the environment conditions can be considered unvarying.

Accordingly, modelling and evaluation solutions integrated in the evaluation workflow belong to specific levels (e.g., construction of topology models belongs to the communication level, stochastic dependability modelling and evaluation belong to the application and architecture levels) and handle the phases (e.g., phases identified on the basis of topology changes).

The concept of hierarchical modelling and solution feedback are directly applied when the application-service dependency models are constructed. If the dependability parameters of an application or service are computed by a specific evaluation technique (that considers the semantics of the service and may apply simulation or experimental approaches, as demonstrated in the previous examples) then the internal software architecture of this service is not explored; instead, it is considered as a single entity in the analysis model and the computed parameters are used. This way the results of specific evaluation tasks related to middleware services and applications can be utilized. If there are no such specific evaluation results then a generic dependability model is constructed taking into account the dependencies at lower levels of the hierarchy (i.e., the mapping to lower level services, hardware and communication resources). The UML object model details the components that are needed for the correct operation of the service, and the dependability parameters of the service are computed on the basis of the local parameters of these components.

6.2 Interactions between tools and techniques

The evaluation workflow consists of several internal model processing steps as presented in Figure 12.

Resolving wireless connections

In this step the required wireless connections are identified on the basis of the application-service dependency model and the topology, taking into account the routing among the participants. For instance the wireless network topology is computed on the basis of the output of VanetMobiSim. In each discrete moment of the mobility trace, the set of available wireless connections (i.e., the current topology) can be determined by a threshold detector based on the distances between the nodes. Here we implicitly assume that the evolution of the wireless network topology can be statically determined.

Identification of phases

The time-based decomposition approach means the identification of phases by merging the changes in the wireless topology and in the user behaviour described in the user workflow model. Phases are time intervals in which the topology and the user activities are static. When the phases are identified an aggregated phase model is constructed that contains all information related to the phases: the phases with the corresponding (fixed) time intervals, the running activities, the nodes and the topology (wireless connections between them). The snapshot belonging to a phase (activities, nodes, and connections) is called a *configuration*.

UML based dependability model construction

In this step the conditions of the successful execution of applications are resolved. The corresponding dependability sub-models that represent the failure and recovery processes come either from results of external modelling or evaluation tasks or from architecture based modelling. In this latter case the dependability sub-models are constructed on the basis of the detailed application-service dependency models.

Shortly summarizing, each hardware and software component is assigned a dependability sub-model, which in our case will be a Stochastic Petri Net that represents a generic fault activation process (healthy, erroneous and failure states of the component are distinguished, where failure means a deviation from the correct service of that component). Several component types can be distinguished based on their faulty behaviour at this abstraction level (e.g., stateful or stateless hardware and software components). The sub-models' parameters are the fault occurrence rate, error activation delay and (optionally) recovery delay. Network connections may exhibit fault activation on their own, thus these are assigned fault activation sub-models in a similar way. The "uses the service of" and "is deployed on" relations among the components and connections are assigned error propagation sub-models with propagation probability parameters. The details of these analysis sub-models and the handling of redundancy are described in [29]. When the application dependability model is constructed, the sub-models corresponding to the different component, connection and relation types are taken from a library of analysis subnets (as patterns), parameterized according to the tagged values of the concrete components in the model, and interconnected through interface places automatically [31].

Generation of the phased dependability model

According to our evaluation approach the analysis model is a Multiple Phased System (MPS) model. This is composed of two logically separate Petri nets: the System Net, representing the structure of the system (users, activities, components and their interactions with failure/recovery), and the Phase Net, representing the phase changes.

The Phase Net model is constructed on the basis of the aggregated phase model as a series of places (each one representing a phase) with timed transitions included between the consecutive places (each having a delay equivalent to the time interval of the given phase).

The System Net is a single model built for the whole scenario, representing all components that are used in at least one phase. It is constructed by an automatic model transformation on the basis of the configurations of phases (available in the aggregated phase model). Here the dependability subnets of applications and services (generated in the previous step) are used. The static wireless network topology is modelled as a set of peer-to-peer connections. The subnet modelling a connection represents the failure-recovery process of the wireless link. The configuration of the components and connections varies in time along the phases, which has to be reflected in the System Net. The presence of a component or a connection in a phase is modelled using proper guard expressions

that refer to the marking of the Phase Net. If a component/connection is not part of the configuration then its subnet is "separated" from the other parts of the model: in fact, the guard conditions of the transitions that represent error propagation become false when the marking of the Phase Net corresponds to the given phase (this way these have no effect on the other subnets).

From the point of view of the phased behaviour there are specific aspects that have to be taken into account. The fault activation subnet of a stateful hardware or a continuously running software service is not influenced by the fact whether it is part of the configuration or not, because faults may occur at any time. However, a stateful piece of software that is demand triggered (e.g., an application started by the user) cannot fail if it is not part of the configuration, and every time it becomes part of the configuration it is started in the healthy state. The users are identified by a specific stereotype in the application-service dependency models. The corresponding subnets do not represent failure and repair processes but include the interface places connected by guarded propagation subnets (belonging to the "uses the service of" links) to the subnets representing the applications invoked by the user. The assigned reward expressions are defined in such a way that the reward is accumulated if all relevant applications are available.

Solution of the phased analysis model

The time distribution functions of the timed transitions in the Phase Net and System Net determine the solution method and the tool that can be used. Simulation based solution by Möbius allows general distributions. Efficient analytical solution by the DEEM tool can be utilised if the MPS model is based on Deterministic and Stochastic Petri Nets. In this case transient analysis of MPS models is carried out using Markov Regenerative Processes, which allows the intra-phase processes to be solved in isolation, thus preventing the state space explosion [32].

6.3 Working with the evaluation workflow

Currently the Viatra2 [33] graph transformation framework is used to carry out the model processing steps based on precise metamodels, and the output MPS model is generated automatically. Some steps (e.g., the identification of phases on the basis of the user workflows and the topology models) are implemented as graph transformations, while more simple steps are implemented as imperative functions. At this stage, the implementation is composed of 13 metamodels, 8 graph transformation steps and 27 native functions. Assessment of the workflow focused on the relevant factors that determine the performance and scalability of the internal model processing steps. An "emergency warning" scenario analyzed by the evaluation workflow consisted of two participants and an ambulance car where the ambulance tries to notify the participants. The mobility traces were created by VanetMobiSim taking into account additional cars that can form a network route between the participants and the approaching ambulance. The evaluation addressed the success probability of the scenario based on different ranges of the wireless connection. The MPS model was generated automatically and its solution was carried out by simulation in Möbius. Different mobility traces were generated with different number of cars in the scenario. It turned out that the dominant part of the time needed to run the evaluation workflow was the construction of the analysis models. Its time depended heavily on the number of cars in the scenario (e.g., increasing the number of cars from 3 to 11 in 4 steps the time increased from 50 to 700 seconds). This significant dependency is due to the current implementation of the exploration of the potential routes between the participants in the different phases (as this step is based on graph transformations, the memory need of this exploration step is also a limiting factor that determines the number of cars that can be handled). The increase of the size of the analysis model is linear with respect to the number of cars (900 to 1300 model elements in the above cases). After the generation of the model, the time needed to run the simulation (get solution with 95% confidence level and 10% confidence interval) was below 10 minutes even in the case of the largest model.

7 Final Remarks

The paper presented the quantitative evaluation performed in the HIDENETS project, focusing on the combination of different techniques to master the overall system complexity and quantify end-to-end quality of service measures. The feasibility of such holistic approach has been first illustrated providing two examples of interaction between mobility simulators and analytical models. Both studies are examples of the cross-fertilization interaction among different methods, since we feed system models with parameter values derived through simulations. The paper has also provided an example of an evaluation workflow supporting the holistic approach, where the models at different levels are aggregated and higher level models are generated in an automated way.

Finally, it is worthwhile to mention that such holistic approach and the devised methodologies have a quite general scope and applicability. Their usefulness and usability are not restricted to the car-to-car communication scenarios used in the HIDENETS project as the main example application domain, but they can be adapted and tailored to other contemporary application fields sharing the growing interconnectivity between different infrastructures and their more and more seamless interactions.

Acknowledgment

This work has been partially supported by the European commission in the FP6 research project HIDENETS [3]. The Telecommunications Research Center Vienna (FTW) is supported by the Austrian Government and by the City of Vienna within the competence center program COMET.

References

[1] M. Radimirsch et al. "Use-case scenarios and preliminary reference model". EU FP6 IST project HIDENETS, deliverable D1.1, September 2006. (http://www.hidenets.aau.dk/Public+Deliverables)

[2] C. Jones and B. Randell. "Dependable pervasive systems". University of Newcastle research report CS-TR-839, 2004.

[3] IST-FP6-26979 HIDENETS - HIghly DEpendable ip-based NETworks and Services. (http://www.hidenets.aau.dk/)

[4] M. Kaâniche, P. Lollini, A. Bondavalli, and K. Kanoun. "Modeling the resilience of large and evolving systems". *International Journal of Performability engineering*, vol. 4, n°2, pp. 153-168, 2008.

[5] J-C. Laprie, "From Dependability to resilience". *Supplemental volume of the 38th IEEE/IFIP International Conference on Dependable Systems and Networks, (DSN-2008)*, Anchorage, Alaska, , 2008.

[6] B. Könning et al. "Final evaluation, consolidated results and guidelines". EU FP6 IST project HIDENETS, deliverable D1.3, January 2009. (http://www.hidenets.aau.dk/Public+Deliverables)

[7] P. Lollini, A. Bondavalli et al. "Evaluation methodologies, techniques and tools (final version)". EU FP6 IST project HIDENETS, deliverable D4.1.2, December 2007. (http://www.hidenets.aau.dk/Public+Deliverables)

[8] M. Lanus, L. Yin, and K.. S. Trivedi. "Hierarchical Composition and Aggregation of State-based Availability and Performability Models". In *IEEE Transactions on Reliability*, 52 (1), pp. 44-52, 2003.

[9] A. Klemm, C. Lindemann, and M. Lohmann. "Traffic modeling and characterization for UMTS networks". In *Proc. IEEE Global Telecommunications Conference, GLOBECOM*, vol. 3, pp. 1741-1746, 2001.

[10] T. Israr, M. Woodside, and G. Franks. "Interaction Tree Algorithms to Extract Effective Architecture and Layered Performance Models from Traces". In *Journal of Systems and Software*, Vol 80 (4), pp 474-492, 2007.

[11] J. Arlat, A. Costes, Y. Crouzet, J.-C. Laprie, and D. Powell. "Fault Injection and Dependability Evaluation of Fault-Tolerant Systems". In *IEEE Transactions on Computers*, Vol. 42 (8), pp 913-923, 1993.

[12] DBench - Project Reports section, project short final report – http://www.laas.fr/DBench.

[13] D. Miorandi and E. Altman. "Connectivity in onedimensional ad-hoc networks: A queueing theoretical approach". In *Wireless Networks 12*, 573587, 2006.

[14] Y. Cheng and T. G. Robertazzi. "Critical connectivity phenomena in multihop radio models". In *IEEE Transactions on Communications*, vol. 37(7), 770777, 1989.

[15] T. Spyropoulos, A. Jindal, and K. Psounis. "An Analytical Study of Fundamental Mobility Properties for Encounterbased Protocols". In *International Journal of Autonomous and Adaptive Communications Systems*, vol. 1, N°. 1, pp. 4-40, 2008.

[16] J. Härri, F. Filali, and C. Bonnet. "Mobility Models for Vehicular Ad-Hoc Networks: A Survey and Taxonomy". In *IEEE Communications Surveys & Tutorials*, vol. 11, N°. 4, 2009.

[17] M.-O. Killijian, M. Roy, G. Séverac, and C. Zanon. "Data Backup for Mobile Nodes: A cooperative middleware and experimentation platform". *Workshop on Architecting Dependable Systems*, Supplemental Volume of the 2009 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-2009), Portugal, 2009.

[18] L. Courtès, O. Hamouda, M. Kaâniche, M.-O. Killijian, and D. Powell. "Dependability Evaluation of Cooperative Backup Strategies for Mobile Devices". In *Proc the 13th IEEE Int. Symp. On Pacific Rim Dependable Computing (PRDC-07)*, 2007.

[19] L. Xu, V. Bohossain, J. Bruck, and D. G. Wagner. "Low Density MDS Codes and Factors of Complete Graphs". In *IEEE Transactions on Information Theory*, 45(1), November 1999, pp. 1817–1826.

[20] O. Hamouda, M. Kaâniche, J-G. Rasmussen, E. Matthiesen Møller, and H-P. Schwefel. "Connectivity dynamics in vehicular freeway scenarios". The *2nd IEEE International Workshop on ITS for an Ubiquitous ROADS (UbiROADS'09)* co-located with IEEE Global Information Infrastructure Symposium (GIIS '09), Hammamet, Tunisia, 22-26 June, 2009.

[21] A. Bondavalli, P. Lollini, and L. Montecchi. "Analysis of User Perceived QoS in Ubiquitous UMTS Environments Subject to Faults". In *Software Technologies for Embedded and Ubiquitous Systems,* LNCS, Springer Berlin / Heidelberg, Volume 5287/2008, Pages 186-197, 2008.

[22] W. H. Sanders and J. F. Meyer. "Stochastic activity networks: Formal definitions and concepts". In *Lectures on Formal Methods and Performance Analysis*, volume 2090 of LNCS, pages 315–343. Springer Verlag, 2001.

[23] A. Bondavalli, P. Lollini, and L. Montecchi. "QoS Perceived by Users of Ubiquitous UMTS: Compositional Models and Thorough Analysis". In *Journal of Software, Special issue on Selected Papers of The 6th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, Volume 4, Issue 7, pp. 675-685, 2009.

[24] D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders. "Möbius: An extensible tool for performance and dependability modeling". In *Proc. 11th Int. Conf. TOOLS 2000*, LNCS, pp 332-336, Springer, Berlin, 2000.

[25] M. Nawrocki, H. Aghvami, and M. Dohler. "Understanding UMTS Radio Network Modelling, Planning and Automated Optimisation: Theory and Practice". John Wiley & Sons, 2006.

[26] H. Holma and A. Toskala. "WCDMA for UMTS: Radio Access for Third Generation Mobile Communications". New York, NY, USA: John Wiley. & Sons, Inc., 2001.

[27] M. Kovacs, P. Lollini, I. Majzik, and A. Bondavalli. "An integrated framework for the dependability evaluation of distributed mobile applications". In *Proc. of the RISE/EFTS Joint International Workshop on Software Engineering for REsilieNt systEms (SERENE 2008)*, pages 29-38, Newcastle upon Tyne, UK, November 17-19, 2008.

[28] A. Nickelsen, and H.-P. Schwefel. "Emulation of Wireless Multi-Hop Topologies with Online Mobility Simulations". *International Journal on Advances in Telecommunications,* Vol. 2, N° 1, pp. 27-36, June 2009

[29] I. Majzik, A. Pataricza, and A. Bondavalli. "Stochastic Dependability analysis of System Architecture Based on UML models". In R. de Lemos, C. Gacek and A. Romanovsky (eds.): *Architecting Dependable Systems*, LNCS-2667, pp 219-244, Springer Verlag, Berlin, 2003.

[30] A. Bondavalli, I. Mura, S. Chiaradonna, R. Filippi, S. Poli, and F.Sandrini. "DEEM: a tool for the dependability modeling and evaluation of multiple phased systems". In *Proc. Int. Conf. on Dependable Systems and Networks,* pp 231-236, IEEE, 2000.

[31] I. Majzik, P. Domokos, and M. Magyar. "Tool-supported Dependability Evaluation of Redundant Architectures in Computer Based Control Systems". In *Proc. FORMS/FORMAT 2007*, Braunschweig, Germany, pp 342-352. GZVB, 2007.

[32] I. Mura and A. Bondavalli. "Markov Regenerative Stochastic Petri Nets to Model and Evaluate the Dependability of Phased Missions". *IEEE Transactions on Computers*, 50(12):1337-1351, 2001.

[33] A. Balogh and D. Varró. "Advanced Model Transformation Language Constructs in the VIATRA2 Framework". In *Proc. ACM Symposium on Applied Computing*, Dijon, France, (SAC 2006), pp 1280-1287, ACM Press, April 2006.