Analysis and Evaluation of the German Toll System using a Holistic Executable Specification

Dr. Bernd Pfitzinger  
FOM Hochschule für Oekonomie & Management, Bismarckstraße 107, 10625 Berlin, Germany  
bernd.pfitzinger@gmail.com

Dr. Tommy Baumann  
Andato GmbH & Co. KG, Ehrenbergstraße 11, DE-98693 Ilmenau, Germany  
tommy.baumann@andato.com

Dr. Thomas Jestädt  
Toll Collect GmbH, Lankstrasse 4, 10785 Berlin, Germany  
thomas.jestaedt@toll-collect.de

Abstract

The European Electronic Toll Service (EETS, [1]) allows different possibilities for the overall system design of interoperable toll systems. We study the effect of the overall system design on the operational costs using a model based, holistic, executable specification. The model is set up to give an accurate description of the interdependences as seen in the German toll system. Two different scenarios are simulated and compared. The first scenario simulates the on-board toll calculation (as used in the German toll system), the second scenario corresponds to backend toll processing. The simulation gives an accurate figure for the operational costs associated with this design decision. A comparison of the cost of financing (of not yet processed toll fees) with the cost of communications is derived from the simulation results and allows choosing the right system architecture and system parameterization according to minimal operational costs.

1. Introduction

The provisioning of complex and cost efficient services typically involves a diverse set of integrated technical (IT) solutions. However, the behavior of the service management organizations, processes and people interacting with the service influences the further development of the service offering in an evolutionary way [2] [3]. The introduction of feedback quickly leads to a seemingly unpredictable system behavior; even small systems develop chaotic behavior [4]. We describe the usage of a simulation using a holistic executable specification of the underlying technical systems to predict the effects of planned changes or to actively optimize the existing service.

As example we use the IT systems of Toll Collect GmbH – the provider of the German electronic toll for heavy trucks. This technical system is able to automatically collect the electronic toll using an on-board-unit (OBU) installed in each truck\(^1\). Currently there are more than 650,000 OBUs deployed, each containing a Global Navigation Satellite System (GNSS) receiver, a GSM modem for communication with the Toll Collect data center and the ability to determine the toll charge due according to the up-to-date map of chargeable roads and the current toll tariffs. Being deployed in trucks, the OBUs are exposed to all sorts of adverse conditions, e.g. mechanical stress, heat, cold, moisture, electrical stress, electromagnetic interference, arbitrary power cycles, unpredictable loss of communications and sometimes a low bandwidth (and high latency) data connection and constrained local computational resources.

Beyond the primary business process of collecting the electronic toll, the system implements a number of supporting processes, e.g. the modeling of changes to the road network, the deployment of updates (to the geo data, the toll tariffs or the OBU software) and of course the more common processes and systems of backend data center processing (e.g. billing, customer relationship management, document management, data analysis and data mining).

Even though the overall system might be considered as almost static in the Toll Collect example, there have been more than 10 major changes (releases) and more than 1,000 medium-sized changes implemented in the past three years. Especially due to the very high level of accuracy needed (with an error rate of less than one in 400 in the automatic toll system [5]) it is vitally important for the system operator to ascertain the effects of changes (e.g. to the software, the configuration, but also of the usage pattern by the truck drivers) as early as possible. Therefore system simulations are used in addition to functional and operational tests to predict the system behavior. In that

\(^{1}\) An alternate mode of operations is available which does not require the installation of equipment in trucks but rather relies on manual booking.
way it is possible to determine the effects of changes not only in the short term but also for several weeks of operations as well as conditions beyond the systems’ specification.

In the context of the EETS directive [1] the different (and technically distinct) European toll systems will evolve towards interoperable systems using GNSS, GSM-GPRS and 5,8 MHz microwave technology. Accordingly the business processes will change, e.g. allowing contractual interoperability for road-users. These changes to the toll systems and its usage pattern necessitate a detailed system simulation to determine the optimal future mode of operations.

A GNSS-based electronic toll system consists of three distinct parts: the on-board-unit, the mobile data communication network and the backend processing in a central data center.

In the German example, every truck participating in the automatic electronic toll collection is fitted with an on-board-unit that automatically calculates the toll charges due according to the current toll tariffs and the geo data related to the toll roads. The OBU transmits the amount due periodically to the backend data center. The OBU uses a number of parameters to determine the optimal instant to transmit the toll charges due, e.g. to limit the total amount or the maximum time between transmissions. These parameters can be configured and updated over-the-air. In the same way regular updates of the geo data, tariffs and the software are distributed over-the-air. All communications between the OBU and the data center are encrypted and authorized in several stages before reaching their final destination – a dedicated high-availability data entry and validation system in the data center.

The mobile data communication is based on the German GSM mobile network operators, providing a nationwide GSM coverage and a closed user group for the Toll Collect system. Resource constraints are encountered at various points within the mobile networks, e.g. quality-of-service prioritization and limited dedicated bandwidth on the technical level. Additional application-level protocols between the OBU and the data entry and validation system in the data center ensure a semantically correct data flow in the correct sequence even when outages (of the mobile network or OBU power cycles) occur.

For the purpose of this article the model of the backend data center consists only of the receiving data entry and validation systems and the systems providing updates to geo data, tariffs and OBU software. The systems for authentication and data encryption add a considerable complexity that is fully included in our simulation model. The further data center processing (e.g. billing, accounting, invoicing etc.) is not included in the model.

We present our results in the following chapters starting with a description of the modeling and simulation framework used, followed by the analysis and evaluation of the simulation, a brief outlook on further applications in the Toll Collect example and a brief conclusion.

2. Modeling and Simulation Framework

To specify and evaluate the toll system, a simulation-driven design approach [6] was selected. The approach is characterized by applying modeling and simulation technologies already in the early design stages [7] [8], when most of the important design decisions are made. It allows validating and optimizing the overall system architecture in the specification phase to avoid integration issues in the later implementation phase [9]. By doing so specification speed and quality is being increased strongly while the system/product uncertainty is being decreased. It is noteworthy that simulation-driven design not only refers to the system under design but also to the surrounding design process. Thereby the process is captured in form of an executable specification which allows automating design steps like architecture optimization, validation against operational scenarios, and tracking of design decisions. Simulation-driven design is supported by the system design tool MSArchitect [10] which we applied to model the toll system. The model execution was accomplished by using the Ptolemy simulation kernel [11].

Figure 1: Solution Framework
2.1. Framework Overview

In a first step we laid out a solution framework consisting of three components. The first component is a model based holistic executable specification (HES) of the toll system describing the functional, architectural and environmental system properties including dynamic interactions between the subsystems. This model is controlled by a second framework component, the graphical user interface (GUI). It controls the parameterization and execution of the HES. Each parameterization includes information like run length, vehicle scenario, and charge model and is called a mission. Such missions can be defined, stored and loaded by means of the GUI. The third framework component is a model based scenario generator (SG). It generates operational vehicle fleet scenarios for the HES, based on historic statistical data. Figure 1 gives an overview over the architecture of the solution framework including the collaborations between the components.

To ensure a high quality and acceptance of our solution framework, it was validated against the real toll system. This was done by comparing input and output data sets between the models and the real system (e.g. average number of transmitted messages) and also by performing source code inspection workshops [12] with system/subsystem specialists.

2.2. Holistic Executable Specification

The core of the simulation framework is the HES of the toll system shown in Figure 2. It is implemented using hierarchical executable composite structure diagrams, whereby the actual functionality is captured in form of C++ source code, the so called atomics (see Figure 3). The model execution is controlled by a discrete event (DE) [13] scheduler. The block “Vehicle Fleet” is responsible to initialize the pool of up to 1.000.000 vehicle objects (OBUs) based on pre-generated scenario data. Each of these OBU objects contains information about the parameterization as well as the internal state of the OBU. In that way we accurately model a fleet of trucks with statistically realistic driving patterns and OBUs with different hardware and software versions and configurations. The realistic model of the vehicle fleet includes arbitrary “power cycles”, i.e. interrupted and aborted transmissions leading to failed data transmissions (that are recovered by application-layer protocols).

Starting from here the automatic communication with the central data center via status messages and data collection messages changes the internal state of the OBU (and of the central IT systems and possibly also of the mobile data network). The block “Mobile Operator” implements provider specific data transmission properties at transport level (e.g. bandwidth and latency) as well as resource constraints (e.g. number of simultaneous connections, quality of service prioritization). In the example discussed we added the ability to collect usage data to facilitate the evaluation of different mobile network cost models.

```c++
inline void PDPAllocate::run()
{
    double arrivalTime = getCurrentTime().toDouble();
    TcmessageRef tcmessage =
        Tcmessage_in.receiveToken( ACCESS_READ );
    LkwRef lkw = tcmessage->lkw;

    Verbindungsstatus_mfbRefArray&
    connectionstate_mfbVector( 
        Connectionstate_mfbVector.getValue().get
        <Verbindungsstatus_mfbRefArray>() );

    Verbindungsstatus_mfbRef& state_mfb
        ( connectionstate_mfbVector[ lkw->_obuid ] );

    // break down existing PDP contexts
    if (state_mfb->_pdp_context)
        FreePDP_out.send( 1 );

    // allocate PDP context
    state_mfb->_pdp_context = 1;

    // notice pdp_context_reservationtime
    state_mfb->_pdp_context_reservationtime =
        arrivalTime;
    Tcmessage_out.send( tcmessage );
}
```

Figure 2: Holistic Executable Specification (HES)

Figure 3: Abstract of the atomic “PDPAllocate”, containing the actual functionality within the hierarchical executable composite structure diagrams
The block “Central System” describes the terminating systems of the central data center of the toll charger including the systems for authorizing data transmissions and validating and acknowledging vehicle data. The setup follows the high-availability approach used for the Toll Collect data center and models all systems involved in their realistic configuration, e. g. firewalls, proxy servers, database and application servers each with their respective connections and their individual resource constraints (e. g. number of parallel connections).

2.3. Scenario Generator

The first part of the HES is initialized by a vehicle fleet scenario consisting of a large number of (individual) OBUs with statistically valid driving patterns (including average distance driven, distribution according to the day of week and the hour, frequency of entering and leaving toll roads, probability of power cycles etc.). These scenarios are pre-generated by the SG and rely on statistical data taken from the real world toll system. Currently we use data from the calendar weeks 49/2010 to 11/2011. Figure 4 shows the sequential generation process of the SG. In a first step the block “Clock” triggers the block “Initialize Vehicle” several times to create and initialize vehicle objects. Thereby the considered period of time and the amount of vehicles is adjustable. Each vehicle object gets randomly assigned (again following the real-world distribution) a device class (e.g. hardware and software type), origin, mobile network provider, and vehicle class (e.g. Six-Wheeler). In addition the vehicle objects get assigned a current version and validness of their software model, toll pricing model, and map of chargeable roads. Next the vehicles get assigned their individual driving patterns as a list of active weeks, their origin and within each of the active weeks a list of active days. Starting from here each active day gets assigned multiple activation intervals describing routes driven (at an average length of 86 km per tollable route [14]).

While driving these routes toll information needs to be collected for toll charging. For this purpose each route is partitioned into multiple road segments each generating an accounting fragment when used by a truck. A fragment is generated about every 5 minutes while the truck remains on the toll road. The simulation includes additional toll free roads (about 15% of the length of the toll road) before and after the toll road. The generation process ends by serializing the data of all vehicle objects into files in preparation for the data transmission to the data center. Figure 5 shows an example about how the generated scenario file can be interpreted. It pictures the driven distance in kilometers of the whole vehicle fleet over a time period of 4 weeks in an interval of 15 minutes. The typical overall driving pattern can easily been seen: Light traffic during the night time and very low truck traffic on weekends (due to the Sunday truck ban on German highways, §30 StVO [25]).

3. Analysis and Evaluation

The depicted solution framework allows analyzing, evaluating, and optimizing functional and architectural system properties. Such properties include operational costs [15], architecture scaling issues, locating architectural bottle necks, testing new system functionality, testing new charging models, and testing new operational scenarios (bigger vehicle fleets). To demonstrate the power of simulations by holistic execution specification we restricted ourselves to one specific example: Following the simulation of [16] we use our simulation to focus solely on comparing the monetary cost of running the toll charging system in either one of the following two scenarios:

A) The toll charging system calculates the toll charge on the OBU and transmits the amount due only after a given threshold is reached.
B) The toll charging system transmits any amount due as soon as possible.

In scenario A we deliberately choose the threshold so that very few transmissions occur whereas in scenario B data is transmitted as soon as possible. Of
course these two models are an example of two extreme configurations with a continuum of possible (and even much more realistic) scenarios in between. In our example there remains a simple trade-off between two expenses: On the one hand frequent data transmissions will incur higher costs and on the other hand the toll charges left in the OBUs can be treated like a credit given to the shipping companies.

3.1 Cost Model

The simulation uses two different cost models, one for the cost of financing (of not yet processed toll charges where we assume a daily bank transfer of all charges collected so far) and another model for the communication costs. In both cases we take the liberty to use publicly available data in the construction of the cost model.

The cost of financing is simply determined by looking up the current bond rate of one of the shareholders of Toll Collect GmbH, e.g. for Deutsche Telekom AG which is currently at 3.74% [17]. In the simulation we calculate once daily the total amount of toll charges due that remain in the OBU and are therefore an open credit to the truck companies.

It is much more difficult to choose an appropriate model for the communication cost. Although there are many consumer contracts available by different network operators, contracts for large-scale machine-machine communication networks are not available in the public domain. For our example we construct a simple cost model using current public information. Unfortunately it is not yet common for mobile network operators to publish a simple metric (comparable to the ARPU metric) for non-voice contracts. Researching the annual reports of the German mobile network operators there even exist completely different ways of measuring non-voice performance. E.g. Telefonica O2 publishes the number of data connections in their annual report [18], but does not mention the amount of data transmitted or the revenues generated by non-voice contracts.

We choose to establish a cost model based on data transmitted (i.e. without any monthly base fees) combining public data of Deutsche Telekom AG [19] and the German telephone network regulator [20] for the year 2010 (see table 1). The annual report of Deutsche Telekom AG lists the non-voice percentage of the ARPU and the monthly ARPU. Together with the market share of the D1 network [21] and the estimated total amount of data (in petabyte) transmitted in the German mobile networks we propose the average price per GB for mobile data transfer in 2010 as 96.1€ (compared to 154 € in 2009 [22]).

<table>
<thead>
<tr>
<th></th>
<th>2009</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARPU</td>
<td>15 €</td>
<td>16 €</td>
</tr>
<tr>
<td>non-voice ARPU</td>
<td>26%</td>
<td>30%</td>
</tr>
<tr>
<td>data volume</td>
<td>33 PB</td>
<td>65 PB</td>
</tr>
<tr>
<td>network users</td>
<td>39,1M</td>
<td>34,7M</td>
</tr>
<tr>
<td>market share</td>
<td>36%</td>
<td>32%</td>
</tr>
</tbody>
</table>

This price per GB is certainly a lower bound in the case of an electronic toll collection system since the bulk of mobile data is transmitted in large batches over fast 3G networks that are typically neither available along toll roads nor used in the current OBU generation (which relies on GSM 2G and 2.5G networks). In the context of a European Electronic Toll Service [23] the toll charger covering more than one country will also incur considerable additional roaming charges, e.g. an additional 800 € / GB (wholesale charge in 2010 by the operator of a visited network as limited by EU regulation [24]).

3.2 Simulation

Based on the cost model above two scenarios for the HES were defined and stored as missions (see Figure 1). These scenarios are mostly equal. They rely on the same vehicle fleet (60,000 trucks over 3 months, i.e. 10% of the size of the real-world vehicle fleet), have the same timing behavior, and use equal resource constraints across the simulation model. The difference between the two scenarios is a single (rather technical) parameter “GPRS-Limit” used to control the amount of toll charges collected by an OBU before the transmission to the data center is triggered. In scenario A the parameter is set to 1024 bytes, in scenario B to 0 bytes. The latter choice forces the transmission of a message every time a section of a toll highway needs to be billed; the first setting allows the accumulation of toll charges and status messages to occur before transmission. In effect this leads to infrequent data transmissions.

The HES was executed for both scenarios with a run length of 7,776,000 time steps (implemented as double in C++). The time steps are interpreted as seconds and represent 3 months of operation in reality. Figure 6 and Figure 7 show an extract of the simulation results for 31 days. The x-axis of the graph stands for the time in days and the y-axis for the...
amount of sent GPRS packets (green) or not yet processed toll fees (red).

Figure 6: Extract of scenario A for 31 (simulated) days

Figure 7: Extract of scenario B for 31 (simulated) days

Both scenarios show a very pronounced driving pattern, e.g. that the toll fees are lower over the weekend (due to the Sunday traffic ban for heavy trucks on German motorways) and higher at weekdays (day one means Monday). Additionally it can be seen that in scenario A the amount of sent GPRS packets is lower than in scenario B. As expected, not yet processed toll fees do not exist in scenario B due to the instantaneous transmission of information.

Due to the interconnectedness of the toll system on the one hand and the required degree precision on the other hand (consideration of dynamic coupling effects between subsystems) the modeled HES is relatively complex. For this reason the simulation performance is the crucial factor to enable a timely performing of analysis, evaluation, and optimization tasks. Table 2 summarizes the utilized CPU time in executing the HES by using the DE scheduler of Ptolemy. Currently a new parallel discrete event simulation (PDES) kernel is in development for MSArchitect which will be applied in the next project phase to increase simulation speed. First tests showed the feasibility of using the whole vehicle fleet in simulations without resorting to any scale factors.

<table>
<thead>
<tr>
<th>fleet size</th>
<th>simulated time period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 week</td>
</tr>
<tr>
<td>600</td>
<td>0,9</td>
</tr>
<tr>
<td>6.000</td>
<td>39</td>
</tr>
<tr>
<td>60.000</td>
<td>18.074</td>
</tr>
<tr>
<td>600.000</td>
<td>3,5 M</td>
</tr>
</tbody>
</table>

Table 2: HES simulation performance (in seconds) depending on vehicle fleet size and simulated days

3.3 Simulation Results

Regarding the results of the second simulated month we calculate for the operational scenarios A and B the cost of financing and the communications cost as discussed in section 3.1 (i.e. with a minimum packet size of 1kB and a purely linear cost model based on packet usage). As expected, scenario B yields zero additional cost of financing, since any amount due is transmitted as soon as possible for backend processing and billing. However, the amount of data transmitted is more than 15 times as large in scenario B as in scenario A. The cost of financing in scenario A is roughly 3 times the communications cost. In total the comparison of the scenarios A and B for communications costs and cost of financing favors scenario A with a rate of 1:4.

4. Results discussion and outlook

Of course the comparison restricted to purely monetary aspects gives a distorted view of the overall system operations. The next step will include minor additional effects (from the systems’ point of view) that might incur additional costs, e.g. the possibility of OBUs using roaming for data communications (e.g. on toll roads close to the border) or the possibility of lost toll charges (e.g. due to hardware defects or bankruptcy of shipping companies).

Beyond purely financial results we use the simulation to predict the systems’ behavior in various conditions – ranging from standard operations to software and data updates or operating conditions (e.g. error recovery in case of equipment failure). Therefore we use the simulation in different scenarios to collect technical metrics; e.g. the load on certain components (e.g. bandwidth, response time, number of simultaneous connections). As a result it is a simple task to run pre-defined scenarios for given changes of
the configuration or the simulation model (either to validate the effects of a planned change or to optimize the system design). The use of a simulation is in many ways the only tool available to predict operational stability and the capability to recover from conditions beyond specification.

5. Summary

Using the simple cost model, we have shown the feasibility of optimizing the system configuration e.g. to yield optimal operational costs. Of course, any system parameter can be subjected to optimization and operational costs are only one of many possible optimization criteria (other criteria could be peak load on certain components, fault tolerance, error rate, deployment rate of software updates, etc.). Regarding cost models the simulation allows to either configure the system optimally for a given cost model or to choose reasonable contract options to fit the system behavior (e.g. it might be mutually beneficial to transmit data preferentially in off-peak hours or limit the frequency of data accesses).

Using our simulation we have demonstrated that a state-of-the-art automatic toll system should transmit toll charges rarely since the cost of mobile data connections is still very high – even within a single operators’ network.

5. References

[17] bond DE000A0T5X07 on 07-Jun-11
[22] Deutsche Telekom AG, “Q3/10 Results Presentation”, Bonn, 04-Nov-2010