Real-time Service Accounting

M. van Le, B.J.F. van Beijnum, B.L. de Goede
Department of Computer Science
University of Twente
Enschede, The Netherlands
{le,beijnum,goede}@cs.utwente.nl

***Abstract*—Offering telematics services toward the end-users involves inter-domain real-time service provisioning, it therefore can also involves inter-domain real-time service accounting. Recognizing the increasing complexity of accounting services due to dynamic service usage behavior of the end-users, the paper addresses a way to first, decompose end-to-end service into accountable service components and second, apply formal modeling and analysis to service component accounting management.**

Keywords—service accounting, service modeling, formal modeling and analysis

I. Introduction

The novel trends of offering telematics services today involves the participation of multiple service providers. Moreover, current business development indicates strong converging process of service providers under the umbrellas of service brokers [1, 2]. This means, in new business scenarios service brokers are capable to make a selection of service components offered by different service providers such as: access provider, connectivity provider and content provider, in order to provide an end-to-end service based on the user's demand and location. In new end-user/provider relationship, where neither access providers nor content providers "own" the customers, accounting management models are required to be real-time, flexible and open.

The remainder of this paper is organized as follows: section 2 addresses current accounting challenges; section 3 presents the principle of accounting management upon distributed service components; section 4 discusses a service scenario and it's possible accounting model; section 5 illustrates an application of formal modeling and analysis to accounting management; section 6 provides conclusions and future work.

II. Real-time Accounting Challenges

Traditional accounting services are usually carried out in batch mode, driven by periodic collection of resource usage information. These accounting services are not designed to support a variety of transaction records reflecting a wide range of customer behaviors.

By the introduction of telematics services on broadband mobile access technologies such as GPRS and future UMTS, accounting systems are required to handle a multitude of complexity due to dynamic service usage and the mixture of simultaneous service provisioning.

Parallel provision of multiple services to a single end-terminal is one of the technical challenges to account for. For example: both voice connectivity and file transfer services can be provisioned at the same time on a mobile device [3]. Consequently, accounting management is required to take care of two separate service components in order to produce an itemized bill. Furthermore, the development of seamless inter-domain service provisioning is pushing accounting management to another limit. If one assumes that inter-domain seamlessness is possible, accounting information among the involved actors need to be interchanged in such a way that accounting approaches its "real-time" significance. Accounting information must be made available to the user right after a service session or even during a service session. To this extend, efficient accounting record format and quick handling of format diversity are of importance.

Another challenge is how to "convince" a service provider to participate in a service session when "off-line" contracts or settlements do not exist. Here, accounting management needs to support dynamic and temporal nature of service usage, where accounting must be followed by payment for a single service component. For example: one-time usage of local weather forecast in a foreign country provided by a local content provider.

The first step toward the providing of "near" real-time accounting information in the retail telecommunication market is prepaid accounting. It has been so far a big success in terms of real-time thanks to the parallel processes of credential monitoring and service usage metering. The major advantage of prepaid being the financial risk reduction for both users and service providers [4].

However, there still exist many research challenges within prepaid accounting when roaming is involved. Take GSM service for example, even though it supports inter-domain roaming, delay still exists up to thirty-six hours to exchange accounting information using Transferred Account Procedure (TAP) among the involved service providers [5].

In order to respond to the pressing needs and requirements on real-time accounting management, following research challenges (among others) need to be overcome:

1) Multi-mode accounting: Accounting systems must support multiple accounting modes, including duration, volume and value simultaneously.

0-7803-7658-7/02/$17.00 (C) 2002 IEEE

2) Flexible accounting: Accounting systems must enable: variable price scheme, service component creation and service component deletion during a service session.

3) Inter-domain real-time accounting: Accounting systems must enable the user to retrieve personal accounting information and to view online customized bill during and immediately after each service session, independently from service usage location.

4) Temporal accounting: Accounting systems must enable the user to obtain temporal services, provided by any service provider who decides to participate in a service session.

III. SERVICE COMPONENT ACCOUNTING

According to Open Service Component Architecture Framework (OSCAF), an end-to-end service session can be decomposed into four service component classes, namely: Connectivity Service Components, Network Service Components, Enabling Service Components and Application Service Components. The reasoning based on OSCAF about end-to-end service session management in terms of (distributed) service components provides many advantages in a user-centric and dynamic service environment [6, 7].

First, integral service session management can take care of overall end-to-end service provisioning. Service session management should be capable to create service sessions from a selection of available service components. Further, it can ensure the seamlessness and transparency of a service session provisioning.

Second, service component management is delegated to the level of service provider that participates in the session. On doing so, end-to-end service management is divided into operational portions, which is less complicated to handle. This means that each involved service provider is responsible for the appropriate delivery of its own service component.

Third, when considering the engagement of a service provider in a session as temporal action (without off-line contract and/or long term commitment), the service provider expects a return of financial compensation for the associated service component usage. The decomposition of distinct service components favors the accountability of value-added service components.

The *Service Management Decomposition* seen from accounting perspective is illustrated in Fig. 1. The figure suggests an overall Service Session Management (SSM) over the provided service components to ensure end-to-end service quality. Service Component Management (SCM) and Service Accounting Management (SAM) can be carried out at individual service provider's level.

The decomposition of accounting down to the level of individual service component is to enable temporal service usage. Furthermore, the separation of SCM and SAM makes it possible to consider accounting as a value-added service that opens the possibility of outsourcing to an independent party (e.g. Mobile Accounting Service Provider).

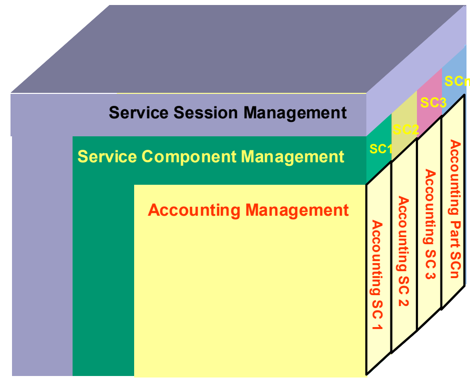


Fig. 1. Service Management Decomposition.

Service life cycles can be defined for the three distinct management areas, namely: *Service Session Life Cycle, Service Component Life Cycle* and *Service Component Accounting Life Cycle*. In a way, these three life cycles acts like a gearing system where the support of one gear to the other is the driving force for the whole system.

The service session life cycle concerns with the integrated management in multi-provider environment to guarantee the end-to-end service provisioning, it consists of:

1) Service Session Invocation: This phase is initiated by the user, requesting a service. The user invokes the Service Session Management by means of a service request.

2) Service Session Negotiation: During this phase, appropriate service components are negotiated, the parameters of negotiation can be specified in a SLA (QoS, Price, etc.). The Service Session Management determines a selection of service components to build up an end-to-end service session.

3) Service Session Creation: Since the Service Session Management is assumed to be capable to negotiate, choose and drop service components offered by the negotiating service providers, this phase confirms the participation of the selected service providers in the provisioning of a service session.

4) Service Session Provisioning, Accounting and Monitoring: Upon the reception of service component confirmations, service instances can be set-up and provisioned. Parallel to this phase, service session accounting is carried out for the provisioned service session. Further, service monitoring is of importance to determine the user's location in order to reconfigure the network for the purpose of seamless handover in a multi-access technologies environment.

5) Service Session Termination: Service termination is the clearing down of a service session. The user can intendedly terminate a session, or the Service Session Management might do so due to running-out user's credit.

The service component life cycle concerns with the operational management to guarantee the provisioning of individual service component, it consists of the following phases:

1) Service Component Invocation: In this phase, the Service Component Management is being invoked for possible service components.

2) Service Component Negotiation: In during this phase, appropriate parameters of negotiation can be specified. The service component management responds with a Price/QoS specification.

3) Service Component Creation: This phase is initiated by the Service Session Management. The Service Component Management starts to configure resources within its domain. During this phase, the service component accounting life cycle can be initiated. The service component life cycle might end here due to a rejection by the service session management.

4) Service Component Provisioning, Accounting and Monitoring: Service Component is actually being provided and monitored. Parallel to this phase, the Service Accounting Life Cycle is carried out for individual provided service component.

5) Service Component Termination: The service component is cleared down. The termination of a service component may occur due to several causes: end of the session, user's initiative, handover, etc.

The service accounting life cycle involves the management of accounting processes for the end-to-end service session as well as individual service component, it consist of:

1) Service Session/Component Declaration: Accounting starts with service component declaration, which is initiated by Service Session Provision, Accounting and Monitoring (see Service Session Life Cycle). This phase describes how the provided service component is to be metered and charged (flat rate, time-based, packet-based, value-based, price, etc.). In addition, the option to indicate frequency of charge during a service session is mandatory to reduce financial risk.

2) Service Session/Component Metering and Charging: This phase measures the usage of delivered Service Component. Metering depends on the service component declaration. Further, prepaid credit needs to be monitor and subtracted from the cost of the provisioned service end-to-end usage. Further, exchange of accounting information among the involved parties is carried out.

3) Accounting Consolidation and Updating: This phase takes place during the "Service Session Termination" (see Service Session Life Cycle) to finalize the accounting life cycle. In a multi-provider environment, consolidation of charging records from different service providers may take place to determine the total cost of the service session, which is followed by user's credit updating.

IV. A Service Scenario

To illustrate a possible application of Service Component Accounting approach as addressed in previous section, we assume a prepaid subscription provided by a service broker. Fig. 2 depicts a service scenario, which consists of a *User* and three service providers located in three different administrative domains: *Service Broker* (SB), *Network Operator* (NO), and *Content Provider* (CP).

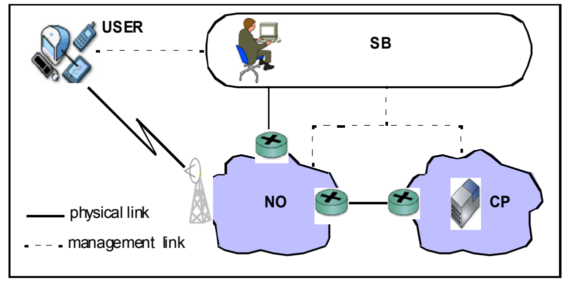


Fig. 2. A Brokerage Service Architecture.

SB maintains the business relations with NO and CP. Therefore, SB is responsible for the appropriate end-to-end service provisioning and accounting of the overall service session. NO is responsible for the negotiated quality of data transmission from CP to the end user. CP takes care of the provisioning of the requested content.

The service scenario presented here deals with a number of accounting challenges. When assuming that each service provider is responsible for the accounting management of the provided service inside its domain, "local" accounting management architecture (e.g. Accounting Server denoted as ASs) is mandatory to enable individual service component accounting. Moreover, accounting information exchange among the involved service providers should be done in such a way that "real-time" accounting is to be realized.

In this example, there is an one-to-one correspondence between the service providers and service component in the service session decomposition. However, it is possible that a service provider can provide more than one constituent service component.

Fig. 3 depicts the coherent existence of three life cycles in terms of message sequence diagram for a successful service provisioning. From the user's perspective, a service session ends at service session termination-phase. From the service providers' perspective, accounting life cycles still continue to exist to enable the consolidation of the total session cost and the updating of user's credit.

V. Formal Modeling And and Analysis

Designing of an accounting architecture is a complicated matter. It is therefore desirable to be able to check the model's correctness and model's behavior at the early state using formal modeling and analysis. In the area of software engineering, these techniques have been applied largely to obtain more understanding of the system 's behavior and to shorten the implementation phase by eliminating undesirable behaviors in advance.

A. Formal Modeling

A simple *Accounting Server* (AS) is shown in Fig. 4 at the level of individual service provider. The model consists of four entities, namely: a *Service Component Manager* (SCM) that manages the provisioning and accounting of a service component; a *Service Component* (SC) that represents the actual provided service; a *Meter* (MT) that simply measure the

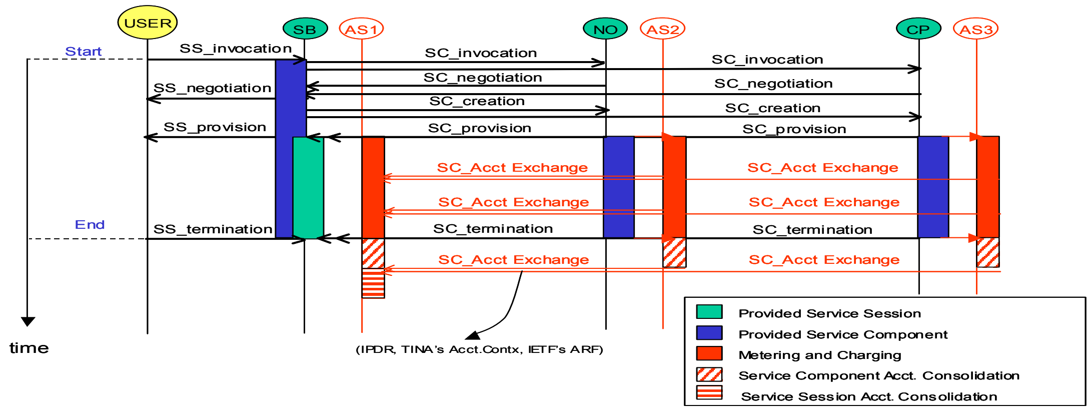


Fig. 3. Coherent Existence of three Life Cycles

service usage; and a *Charger* (CH) that is responsible for the calculation of the actual cost of service usage in real-time.

In order to obtain the overall structural model and behavioral model of the accounting server, the following steps are made:

- *Structural modeling using the Architecture Description Language called Darwin* - This step defines the relationship between the components in a structural model [8, 9].

- *Behavioral modeling using Labeled Transition Systems (LTS)* - In this step, the behavior of accounting server is described accordingly to the accounting life cycle defined in section 3. Each constituting component is described in terms of finite state processes [9] and the synchronization of processes happening inside the three components is a composite LTS, which reflects the overall behavior of the accounting server.

- *Model analysis* - This step verifies the correctness of the accounting server. By stepping through successive events in a process, the desired behavior of a component can be verified. Moreover, a synchronization of processes often results in a deadlock situation. Model checking therefore helps to avoid model ‘s incorrectness.

Followings are examples of behavioral modeling of important entities in the model.

Service Component Manager

1) Service Component Invocation: This phase is assumed to be ready.

2) Service Component Negotiation: This phase is assumed to be ready.

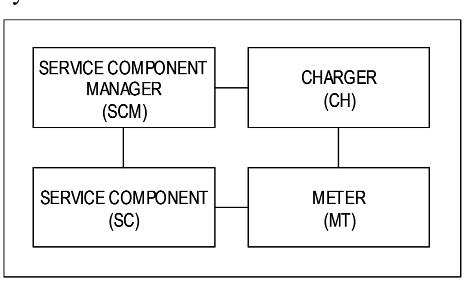


Fig. 4. Coherent Existence of three Life Cycles

3) Service Component Creation: SCM initiates the Service Component Life Cycle by providing CH with price information [p: Price] and frequency of charge.

4) Service Component Provisioning and Monitoring: SCM initiates SC to start provisioning the service component.

5) Service Component Termination: SCM triggers SC to terminate the service component provisioning.

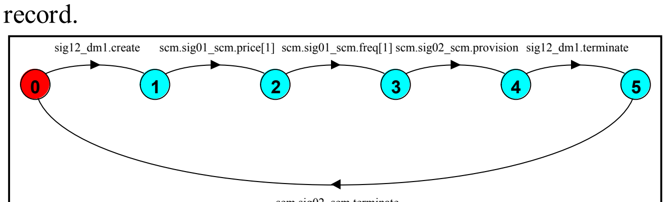
Fig. 5 presents the graphical Labeled Transition System and the textual Finite State Process (FSP) description of SCM accordingly to the service component life cycle.

Charger

1) Service Component Declaration: CH start its process by receiving charging information about service price and frequency of charge indicated by SCM.

2) Service Component Metering and Charging: In this phase, CH can read the meter status[v:1..N] to determine the increasing cost of the ongoing service component. The charging frequency f indicates the threshold based on which charging records bill[i][v][p] must be produced and sent to the Service Broker. For instance, f=2 indicates that charging records must be produced for each two successive service units.

3) Accounting Consolidation and Updating: Upon indication of MT, CH consults MT for the last time to get the final counter’s value, it then calculates the total cost of the component service usage and produces the final charging record.




```
SCM = (sig12_scm.create ->
sig01_scm.price[p:Price] ->
sig01_scm.freq[f:Freq] -> sig02_scm.provision ->
sig12_scm.terminate -> sig02_scm.terminate ->
SCM).
```

Fig. 5. Behavioral Model of the Service Component Manager

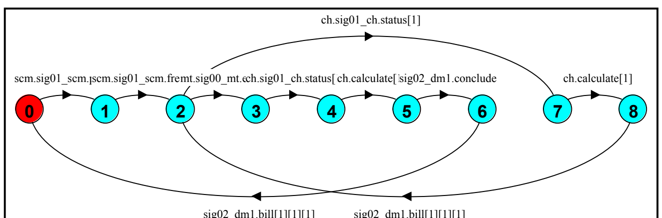
Fig. 6 presents the graphical Labeled Transition System and the textual FSP description of CH accordingly to the service component accounting life cycle.

B. *Safety Property*

An accounting server must ensure a number of safety properties in order to minimize possible financial risk. For instance: one can define a system property to make sure that an accounting process for a certain service component stops when the component is removed from a service session, or one can ensure that a service provisioning stops when prepaid credit reaches zero. Fig. 7 illustrates a safety property, which states that a service provisioning must stop immediately after the credit is empty. The LTS diagram of Fig. 5 reveals that in translating a property process, the compiler automatically generates the transitions to error state. Displaying "[o]" (e.g. credit is empty) twice is a violation of the SAFETY property.

C. *Real-time Relevancy*

The real-time aspect of the presented accounting model is supported by the frequency, by which intermediate charging records must be exchanged between the service providers and the service broker during the service-provisioning phase. The mechanism introduced here looks similar to the Hot Billing Approach [4]. However, our proposed accounting architecture examines the possibility to support inter-domain accounting in real-time and to reduce potential bad debt, which we believe can be minimized by frequential charging record exchange between the involved actors. Having said that, performance issue of the proposed architecture in operational environment is of high importance. We recognize the fact that scalability challenges must be taken into account in order to satisfy one of the design requirements. The formal modeling technique as applied in this work can only provide functional analyses of a model, which is however an essential step in the conceptual study phase.




```
CH =(sig11_ch.price[p:Price] ->
sig11_ch.freq[f:Freq] -> CH[1][p][f]),
CH[i:Bill][p:Price][f:Freq] =
(sig01_ch.status[v:1..N] ->
if ((v<N) && (v-i*f)<0)
then (return ->    CH[i][p][f])
else if ((v<N) && (v-i*f)>0)
then (return -> CH[i][p][f])
else if ((v<N) && (v-i*f)==0)
       then (calculate[v*p] ->
       sig02_ch.bill[i][v][p]-> CH[i+1][p][f])
       else if (v==N) then (calculate[v*p] ->
       sig02_ch.bill[i][v][p] -> CH[i][p][f])
       | sig10_ch.end ->
       sig01_ch.status[v:1..N] -> calculate[v*p]
       -> sig02_ch.conclude ->
       sig02_ch.bill[i][v][p] -> CH).
```

Fig. 6. Behavioral Model of the Charger

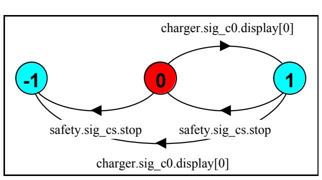


Fig. 7. Accounting Safety Property

VI. Conclusions and Future Work

The reasoning about accounting management on the basis of accountable service components based on OSCAF favors the development of accounting as value-added service on top of the actual end-to-end service provisioning. Further, the paper demonstrates the advantageous application of formal modeling technique and model analysis to accounting management for distributed service components in "near" real-time. Formal modeling enables the definition and the checking of safety properties that are crucial for the accounting system 's security.

Our future work includes the extension of current accounting architecture into multi-domain environment and the impact of user's mobility reflected on accounting management. Further, scalability study of real-time accounting management using a realistic test-bed also takes part of our focus.

Acknowledgment

The authors wish to thank Prof. Dr. G. B. Huitema (KPN Research) for his valuable contribution in the course of this study.

References

[1] Aspiro, http://www.aspiro.com.

[2] Vizzavi, http://www.vizzavi.com.

[3] M Pencen, "Simultaneous voice and data operation for GPRS/EDGE: Class a dual transfer mode", *IEEE Personal Communication Magazine*, vol. 8, pp. 14-29, April 2001.

[4] Y. Lin, M. Chang, "Mobile prepaid services", *IEEE Personal Communication*, vol. 7, pag. 6-14, June 2000.

[5] C. Gullstrand, "Tapping the potential of roaming", http://www.gsmworld.com/using.billing/potential.shtml.

[6] B.L. de Goede, *Operational Management of Telematics Systems and Services*, ISBN: 90-365-1709-5, December 2001.

[7] M. van Le, B.J. van Beijnum, B.L. de Goede: " Formal modeling of service session management", *IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, Santa Barbara - California, Oct. 2002.

[8] K. Ng, et al, "A visual approach to distributed programming", *Tools and Environments for Parallel and Distributed Systems*, A. Zaky and T. Lewis (Eds.), Kluwer Academic Publishers, Feb. 1996.

[9] Dulay, "Darwin reference manual", Dept. of Computing, Imperial College, 1994.

[10] J. Magee, J. Kramer, *Concurrency: State Models & Java Programs,* Wiley, 1999.