CAMEL Service Interaction Detection

I. Atanasov, E. Pencheva

Key Words: *CAMEL; call control; mobility management control; satisfiability; inference algorithm.*

Abstract. *Service interaction manifests itself as a function of services which neither acts exactly as a sum all services nor behaves as expected. In Customized Application for Mobile network Enhanced Logic (CAMEL), services may interact both in call-related and call-unrelated context. A formal approach to detection of CAMEL service interaction in mobile communication system is suggested. The approach is based on CAMEL basic call state models and mobility management model. It takes into account the distributed nature of call control and also allows expression of behavior related to mobility management and to out-band signaling procedures. Description logic is used to formalize the behavior of the system and services. A standard inference algorithm is used to detect feature interaction.*

1. Introduction

Service interaction problem arises after the introduction of Intelligent Network where the number of customized services and features increases.

The service (feature) interaction problem has been intensively studied through the years but still there is no global solution. Common practice of the operators is to avoid any kind of feature interaction. In the design of service control model so that certain interactions cannot occur, the combinations of services that might cause troubles are forbidden. However, restrictions on usage of certain services reduce the flexibility in service creation. Further, the service interactions can never be completely avoided as there are almost always interactions that slip through the networks.

Significant research efforts were made into service interaction detection. Formal model based solutions apply formal reasoning to detect service interaction. The formal models use finite state machines [1,2,3,4], temporal logic [5,6,7,8], process algebra [9,10,11,12], and Petri nets [13,14,15,16]. The applicability of the approach depends on the expression of the expected service's behavior and its implementation in the network.

On one hand, the interactions within a given model can be found after conducting complete analysis which is possible only if the approach is formalized. On the other hand, the exponential increment of models' size with respect to size of the problems might become an issue. Having a reasonable set of service can lead to models to explode in number of their states and each state will need further exploration.

The goal of the paper is to provide description of mobility management and call processing in mobile networks and a straightforward approach to express the behavior of CAMEL services. CAMEL stands for Customized Application for Mobile network Enhanced Logic and it is Intelligent Network for mobile networks [17]. The approach is based on CAMEL basic call state models, attach-detach model, Packet Data Protocol (PDP) context state model and Short Message Service (SMS) state model. We illustrate the capabilities for detection of service interaction for call related services triggered by mobility management events. In comparison with the quoted approached, our approach can also be reformulated to signaling connections that are not call related such as Short Message Service or Internet Protocol (IP) connectivity.

The paper is structured as follows. In Section 2, we briefly discuss previous work connected with our approach to service interaction detection. Section 3 starts with introduction to description logic and then discusses its usage for CAMEL state models description. The approach to service definition and the algorithm for inference of features interaction are presented in Section 4 and Section 5. At last we conclude the paper.

2. Related Work

A number of authors had suggested that feature interaction detection is a satisfiability problem. Starting point for our research was the work of C. Areces, W. Bouma and M. Rijke [18]. The authors use a formal model for the specification of features by means of description logic. Their formal definition of basic telephone system and additional features is based on the basic call processing. Properties of features are formally proved and interactions are detected by means of standard reasoning tasks from description logic.

The authors present a formal language $\mathcal{FI}$ to define concepts related to possible subscriber states, network states and subscriber actions. For example, $ringing_v$ $ringback_u$ and $engaged_u$ are subscriber states representing the ringing terminal with the receiver onhook, the receiver is off and emits a ring back tone, and a connection with another party respectively, while $calling_{uv}$ and $path_{uv}$ are internal states of the network, representing the terminal at v is ringing with u waiting for v to accept the call, and u and v can communicate respectively. First-order logic is used to define terminologies and assertions, and also to reason about them. For example, the following statements connect the observable states of the terminal with ones representing network states

$$calling_{uv} \sqsubseteq ringing_v \sqcap ringback_u \quad u \neq v$$
$$path_{uv} \sqsubseteq engaged_u \sqcap engaged_v.$$

There are statements specifying how a user and the network can change state.

The approach proposes a „good logic" to reason about features and the interaction problem, but its view of the network is simplistic on purpose. In reality, we can not manage the network as if it is a single network element with a basic call process for each call. The call may originate from an exchange and terminate in another one, or even may connect subscribers in different networks.

To provide more realistic model of the network, reflecting the distributed nature of the call control, more detailed representation of the basic call process is needed. The CAMEL splits the basic call process into two parts, called originating basic call state model (O_BCSM) and terminating basic call state model (T_BCSM) [19]. The models describe procedures for initiating and receiving call respectively. In both models, detection points are defined, at which the service logic can be triggered. In addition to basic call state models, CAMEL introduces three new models that allow service logic to control procedures related to mobility management, data session establishment and messaging [19].

Our approach is based on CAMEL call control and mobility management models to describe the behavior of mobile communication system (MCS) in the context of call-related service. The approach possesses more expressive power as it distinguishes between call processing for the originating party and terminating party, and considers mobility management also. Following the same approach it is also possible to express service interactions for data communications and sending and receiving short messages.

3. Description Logic for CAMEL Services

A. Basics of description logic

Consider a domain Δ with fixed names of the concepts and the roles, and given constants forming a triplet $\langle$C, R, A$\rangle$ so that one can define the set of the concepts C, terminologies $\mathcal{T}$ and assertions $\mathcal{A}$(i.e. allowed formulae in TBox and ABox respectively) as follows:

$$\mathcal{BC} \coloneqq \top \mid C \mid \neg\mathcal{BC} \mid \mathcal{BC} \sqcap \mathcal{BC}$$
$$C \coloneqq \mathcal{BC} \mid \forall R.\mathcal{BC} \mid \exists R.\mathcal{BC}$$
$$\mathcal{T} \coloneqq \mathcal{BC} \sqsubseteq C \mid \mathcal{BC} \equiv C$$
$$\mathcal{A} \coloneqq \mathrm{a}{:}C \mid (\mathrm{a,b}){:}R$$

where any basic concept C is in C, any basic relation R is in R, and constants a, b are in A. The pair $\langle$T, A$\rangle$ is *knowledge base* where $\mathrm{T} \subseteq \mathcal{T}$ and $\mathrm{A} \subseteq \mathcal{A}$.

Interpretations of description logics are $\mathcal{I} = (\Delta^I, \cdot^I)$ where Δ^I is a non-empty set and $\cdot^I$ is a mapping of subsets of Δ^I onto the concept names, relations over Δ^I onto role names, and elements of Δ^I onto constants. The interpretations are extended over C as follows:

$$(\top)^I = \Delta^I$$
$$(\bot)^I = \varnothing$$
$$(C \sqcap D)^I = C^I \cap D^I$$
$$(\neg C)^I = \Delta^I \setminus C^I$$
$$(\forall R.C)^I = \{ \mathrm{a} \in \Delta^I \mid \forall \mathrm{b}\ ((\mathrm{a,b}) \in R^I \rightarrow \mathrm{b} \in C^I) \}$$
$$(\exists R.C)^I = \{ \mathrm{a} \in \Delta^I \mid \exists\ \mathrm{b}\ ((\mathrm{a,b}) \in R^I \wedge \mathrm{b} \in C^I) \}$$

In fact, $\forall R.C$ means the whole subset of elements of Δ that are in relation R with the element C. The duality of operator $\forall$ can be expressed in terms of operator $\exists$, as $\forall R.C = \neg\exists R.\neg C$.

The definition of *satisfaction* is intuitive as a relation between interpretations and terminologies or assertions. Thus $\vDash$ is a relation between $\mathcal{I}$ and all formulae supported by $\mathcal{I}$:

$$I \vDash C \sqsubseteq D \text{ iff } C^I \subseteq D^I$$
$$I \vDash C \doteq D \text{ iff } C^I = D^I$$
$$I \vDash \mathrm{a}{:}C \text{ iff } \mathrm{a}^I \in C^I$$
$$I \vDash (\mathrm{a,b}){:}R \text{ iff } (\mathrm{a}^I, \mathrm{b}^I) \in R^I$$

So, for a subset $K \subseteq \mathcal{T} \cup \mathcal{A}$ one may state that $I \vDash K$ iff $I \vDash \varphi{:}\ \forall\varphi \in K$. One refers to a *consequence* if there is a knowledge base $\langle$T, A$\rangle$ and formula φ such that $\varphi \in \mathcal{T} \cup \mathcal{A}$ then φ *follows from* $\langle$T, A$\rangle$ i.e. $\langle \mathrm{T, A} \rangle \vDash \varphi$ iff $\forall \mathcal{I}{:}\ \mathcal{I} \vDash \langle \mathrm{T, A} \rangle \Rightarrow \mathcal{I} \vDash \varphi$ where $\Rightarrow$ notes implication. One of the main reasoning tasks in the description logic is to check whether a given formula follows from a given knowledge base.

B. CAMEL models

CAMEL provides models describing the process for initiating calls (O_BCSM) and the process of receiving one (T_BCSM). Both models can trigger CAMEL service logic. Both O_BCSM and T_BCSM are represented as state-transition diagrams that describe the call processing states and transitions between them. The states are named points in call and the transitions are caused by events such as call initiation, call answer, or disconnect. An event can have an associated detection point at which the service logic can be invoked if predefined criteria are met. CAMEL T_BCSM is shown in *figure 1*.

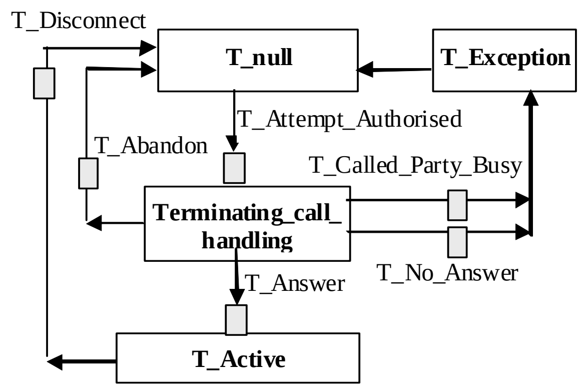

Figure 1. CAMEL T_BCSM

The CAMEL Attach-detach model is defined to enable the service logic to control the GPRS mobility management. This model tracks the mobility management procedures for GPRS and allows the service logic to intervene in them. *Figure 2* shows the CAMEL Attach-detach model.

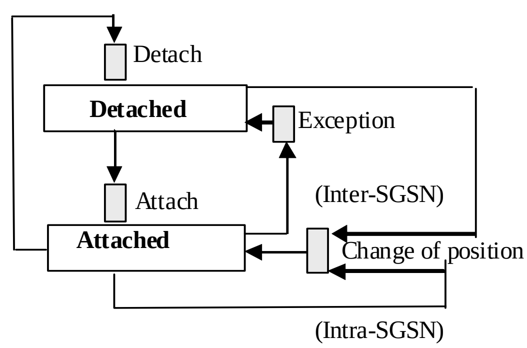

Figure 2. CAMEL Attach-detach model

Although there is no Attach-detach model explicitly defined for circuit switched subscribers, [19] defines a mechanism for service logic notification about events concerning the mobility management in circuit switched domain. The service logic may be triggered on events related to location update and attach, detach procedures.

Table 1. Atomic concepts related to call and mobility management states

Basic call states	Description
T_Null	There is no incoming call. The authority to route the call to the terminating party is verified.
T_Call_Handling	Terminating resource is informed of incoming call. Routing address and call type are interpreted. The next route or terminating access is selected. The Call Forwarding supplementary service is invoked if necessary. The terminating party is alerted.
T_Active	Connection is established between originating and terminating party.
T_Exception	Default handling of the exception condition is being provided in the terminating party.
Attached	The terminal is attached to the network.
Detached	The terminal is detached from the network.
Idle	The subscriber is idle.

Table 2. Atomic roles related to call and Mobility management transitions

Transitions in models	Description
T_Att_Auth	Indication that the terminating CAMEL subscription information is analyzed.
T_Answer	Call is accepted and answered by terminating party.
T_ Called_Party_Busy	Indication that a busy indication is received from the destination exchange or not reachable or call establishment failure event is determined.
T_No_Answer	Indication that an application timer associated with the T_No_Answer detection point expires.
T_Abandon	A disconnect indication is received from the originating party during the call establishment procedure.
T_Disconnect	A disconnect indication is received from the terminating party or from the originating party.
Detach	The mobile terminal performs detach procedure.
Attach	The mobile terminal performs attach procedure.
ChangeOfPosition	The mobile terminal performs location area update.
Exception	The signaling connection fails.

C. Description of CAMEL models

Our approach to definition of atomic concepts is to represent each call state and mobility management state in models as a separate concept as shown in *table 1*. Further we also define the subscriber state.

Each transition in the models is defined as a role since if it has an encountered detection point, the call processing or mobility management will be suspended and the control will be handed over the service logic. The roles and their meaning are shown in *table 2*. The roles represent possible actions of subscribers or default handling by the network.

Our terminology box contains expressions showing the changes in models and statements specifying the relationship between the events that cause the call processing and mobility management (*table 3)*.

We denote by SUBS the set of all subscribers, by TBCSM the states s_i in the T_BSCM model, and by MM the states s in the Attach-detach model. The assertion box contains one statement presenting the initial state for each subscriber:

s_0: $\sqcap_{u \in SUBS}$ (Detached).

We use the statement to express the fact that each subscriber is in exactly one TBCSM state at any moment:

$$\top \sqsubseteq \neg(\sqcup_{s1,s2 \in TBCSM,\ s1 \neq s2}(s1 \sqcap s2)) \sqcap (\sqcup_{s \in TBCSM}\ s).$$

The state of party changes by means of actions defined as

Table 3. TBox representing changes in call and attach-detach models

Statement	Description
$Detached_A \sqsubseteq \exists Attach.Attached_A \sqcap Idle_A$	If the terminal is detached from the network, it may perform attach procedure which results in attached and idle states.
$Attached_A \sqcap Idle_A \sqsubseteq T\_Null$	If the calling party is attached and idle there is no incoming call.
$T\_Null \sqsubseteq \exists T\_Att\_Auth.T\_Call\_Handling$	In case of no incoming call, if the terminating attempt is authorized the result is incoming call handling.
$T\_Call\_Handling \sqcap Detached_B \sqsubseteq \exists T\_Called\_Party\_Busy.T\_Exception$	During incoming call handling if the called party is detached, then a busy indication is received and default handling takes place.
$T\_Call\_Handling \sqcap Idle_B \sqsubseteq \exists T\_Answer.T\_Active \sqcup \exists T\_No\_Answer.T\_Exception \sqcup \exists T\_Abandon.T\_Null$	During incoming call handling if the called party is idle, then she may answer which results in active state, or she may not answer which results in timer expires and default handling, or an indication that the calling party abandons the call may be received which results in state with no incoming call.
$T\_Call\_Handling \sqcap \neg Idle_B \sqsubseteq \exists T\_Called\_Party\_Busy.T\_Exception$	During incoming call handling if the called party is not idle a busy indication is received and default handling takes place.
$T\_Exception \sqsubseteq T\_Null$	The default handling results in state with no incoming call.
$T\_Active \sqsubseteq \exists T\_Disconnect.T\_Null$	During active state, receiving a disconnect indication results in a state with no incoming call.
$Attached \sqsubseteq \exists Detach.Detached$	If the terminal is attached to the network, it may perform detach procedure which results in detached state.
$Detached \sqsubseteq \exists ChangeOfPosition.Attached$	The mobile terminal detaches from the old serving node and attaches to a new one.
$Attached \sqsubseteq \exists ChangeOfPosition.Attached$	The mobile terminal performs intra serving node location area update.
$Attached \sqsubseteq \exists Exception.Detached$	The mobile terminal loses the signaling connection.
$Attached \equiv \neg Detached$	The attached state is the opposite of detached state.

action functions. An action function $Func_{BCSM}$ for a given state corresponds to the possible transitions in the TBCSM. For example, the expression

$$Func_{BCSM}(T\_Call\_Handling)=\{T\_Called\_Party\_Busy\}\cup\{T\_No\_Answer\}\cup\{T\_Answer\}\cup\{T\_Abandon\}$$

means that in case of an incoming call, the terminating party may be busy or not reachable, may not answer at all, may answer or may receive an indication that the originating party abandons the call.

The fact that each party can change the TBCSM state only by means of certain actions is represented by the following statement:

For all $s \in TBCSMs$, and all $R \notin Func_{BCSM}(s)$, $s \sqsubseteq \forall R.s$.

The same is applied to the Attach-detach model where an action function $Func_{MM}$ for a given state corresponds to the possible transitions in the Attach-detach model. For example, the activation functions $Func_{MM}(Detached)=\{Attach\}$, $Func_{MM}(ChangeOfPosition)=\{Detach\}$, and $Func_{MM}(Attach)=\{Detached\}$, describe all possible changes of Detached state.

Then having the statement

For all $s \in MM$, and all $R \notin Func_{MM}(s)$, $s \sqsubseteq \forall R.s$

we can derive $Detached_C \sqsubseteq \forall T\_Call\_Handling_{AB}.Detached_C$.

4. Service Models

A. An approach to definition of services and interaction

Services are modeled by *refinement*.

The definition of refinement is formalized as the refinement operation δ_F, for a given service F, which transforms a given knowledge base K into another knowledge base $\delta_F(K)$. The last is augmented by a set of *activation concepts* which generally are $A_F \subseteq \{ F_u \mid u \in SUB \}$. So, let $N \subseteq A_F$ then K and F *interact on activation* N if $\delta_F(K) \cup \{\top \sqsubseteq F_u \mid F_u \in N\} \cup \{\top \sqsubseteq \neg F_u \mid F_u \in A_F \setminus N\} \models \neg\top$

Let one has different services i.e. F_a, F_b : a≠b. Then $\delta_{Fb}(\delta_{Fa}(K)) \equiv F_a \circ F_b$ and the services under consideration are

Table 4. TBox modified by location changed alerting service

Statement	Description
$C_1[\neg \sqcup_{b\in SUB} LCA_b \sqcap T\_Call\_Handling_{ab} \sqcap Detached_b] \sqsubseteq \exists T\_Called\_Party\_Busy_{ab}.C_2[T\_Exception]$	If the LCA_b service is not activated for the subscriber *b*, and an incoming call from subscriber *a* to *b* takes place, and *b* is not reachable then a busy indication is received and default handling takes place.
$C_1[MCA_b \sqcap T\_Call\_Handling_{ab} \sqcap Detached_b] \sqsubseteq \exists T\_called\_Party\_Busy_{ab}.C_2[T\_Exception \sqcap \neg Notified_a]$	If the MCA_b service is activated for the subscriber *b*, and an incoming call from subscriber *a* to *b* takes place, and *b* is not reachable then a busy indication is received and default handling takes place, and the subscriber *a* is marked as not notified.
$C_3[MCA_b \sqcap Attached_b \sqcap Idle_b] \sqsubseteq \exists Change\_Of\_Position_b.C_4[Attached_b] \sqcap \exists T\_Attem\_Auth.T\_Call\_Handling_b] \sqcap C_5[Notified_b]$	If the LCA_b service is activated for the subscriber *b*, and *b* changes position and *b* is idle then a call is initiated to *b* and a busy indication must not be received.
$LCA_b \sqsubseteq T\_Call\_Handling_b \sqcap Idle_b \sqcap \neg T\_Exception$	If the MCA_b service is activated for the subscriber *b*, and the subscriber *a* is notified, and *b* is attached and idle then a busy indication must not be received.

Table 5. TBox modified by call forwarded unconditional service

Statement	Description
$C_1[\neg \sqcup_{c\in SUB} CFU_{bc} \sqcap Idle_b \sqcap T\_Call\_Handling_{ab}] \sqsubseteq \exists T\_Answer_{ab}.C_2[T\_Active] \sqcup \exists T\_No\_Answer_{ab}.C_2[T\_Exception]$	If the subscriber *b* has not forwarded calls, and subscriber *a* places a call to *b*, and *b* is idle, then *b* may answer and become active or *b* may not answer and the timer expires.
$C_1[CFU_{bc} \sqcap T\_Call\_Handling_{ab} \sqcap Idle_c] \sqsubseteq \exists T\_Answer_{ac}.C_2[Active_{ac}] \sqcup \exists T\_No\_Answer_{ac}.C_2[T\_Exception]$	If the subscriber *b* has forwarded calls to subscriber *c*, and the subscriber *a* places a call to *b*, and *c* is idle then *c* may answer and become active or *c* may not answer and the timer expires.
$C_3[CFU_{bc} \sqcap T\_Call\_Handling_{ab} \sqcap \neg Idle_c] \sqsubseteq \exists T\_Called\_Party\_Busy_{ac}.C_4[T\_Exception]$	If the subscriber *b* has forwarded calls to the subscriber *c*, and the subscriber *a* places a call to *b*, and *c* is not idle then a busy indication is received.
$C_5[CFU_{bc} \sqcap T\_Call\_Handling_{ab} \sqcap Detached_c] \sqsubseteq \exists T\_Called\_Party\_Busy_{ac}.C_6[T\_Exception]$	If the subscriber *b* has forwarded calls to the subscriber *c*, and the subscriber *a* places a call to *b*, and *c* is detached then a busy indication is received.

such that $F_a \circ F_b \equiv F_b \circ F_a$. We use contexts $C[\varphi]$ to define refinements in the knowledge base, where φ is a subformula of any formula ψ.

B. Definition of Missed Call Alerting Service

The Missed Call Alerting (MCA) service notifies the originating party that the terminating party becomes reachable. It is activated for the terminating party in case the party is not reachable. This means that the T_BCSM is applied. In case of an incoming call to detached terminal, the MCA service marks the originating party as not notified, and when the terminating party attaches to the network, the MCA notifies the originating party. The refinement for MCA service is defined by statements shown in *table 4*.

C. Definition of Call Forwarding Unconditional Service

The Call Forwarding Unconditional (CFU) service allows the called subscriber to forward the incoming calls unconditionally to another subscriber. The refinement for CFU service is defined by statements shown in *table 5*.

A possible service interaction occurs when a subscriber *B* which is detached, has forwarded the incoming calls to a subscriber *C* which is also detached. When the subscriber *A* calls to *B*, a busy indication is received and *A* is marked as not notified. When *B* attaches to the network, *A* is notified. Then when *A* places a call to *B*, the call is forwarded to *C* and a busy indication is received.

5. Detection of Service Interaction

A. Tableau method

A knowledge representation system based on description logics is able to perform specific kinds of reasoning. For example, it is important to find out whether a newly defined concept makes sense or it is contradictory. From a logical point of view, a concept makes sense for us if there is some interpretation that satisfies the axioms of T (that is, a model of T) such that the concept denotes a nonempty set in that interpretation. A concept with this property is said to be *satisfiable* with respect to T and *unsatisfiable* otherwise.

Description logics as that for representation of mobile communication system behavior with negation and disjunction can be handled by so-called *tableau-based algorithms*. Instead of directly testing subsumption of concept descriptions, tableau-based algorithms use negation to reduce subsumption to (un)satisfiability of concept descriptions: $C \sqsubseteq D$ if $\neg C \sqcup D$ is unsatisfiable.

We use a tableau method [17] to detect feature interaction.

Table 6. Tableau method

AND: $\dfrac{\langle b \mid p : C \sqcap D\rangle}{\langle b \mid p : C\rangle \quad \langle b \mid p : D\rangle}$	
OR: $\dfrac{\langle b \mid p : C \sqcup D\rangle}{\langle b_M 0 \mid p : C\rangle \quad \langle b_M 1 \mid p : D\rangle}$	b_M maximal for b
SOME: $\dfrac{\langle b \mid p : \exists R.C\rangle}{\langle b \mid pRn : C\rangle}$	pRn new (unless pR exists in the branch)
ALL: $\dfrac{\langle b \mid p : \forall R.C\rangle}{\langle b \mid pRn : C\rangle}$	pRn present in b
KB: $\dfrac{\vdots}{\langle b \mid p : \neg C \sqcup D\rangle}$	with p present in b and $C \sqsubseteq D \in T$

The tableau $t \stackrel{\text{def}}{=} \{ \langle b \mid p\text{: } C \rangle \}$ is a set of prefixed formulae where the prefix of given formula consists of a binary string $b := \varepsilon \mid (1|0)^+$ and a string of alternating names $p := n(Rm)^+$, and C is a concept. Here ε is the empty string, n and m are names of individuals, R stands for the names of roles, and $()^+$ denotes one or more occurrences. Strict or relaxed prefix σ_1 of given string σ_2 can be defined by total ($\sigma_1 \prec \sigma_2$) or partial ($\sigma_1 \preceq \sigma_2$) order. Then b_M is called maximal for b in t if $b \in t \wedge b_M \in t \wedge b_M \preceq b \wedge (\neg\exists b_1 \in t\text{: } b_M \prec b_1 \wedge b_1 \preceq b)$.

B. Detection of MCA and CFU interaction

The tableau algorithm for detecting interactions between MCA and CFU services proceeds as follows:

1. Applying AND to the start formula $\langle \varepsilon \mid s_0\text{:} \sqcap_{u \in SUB}$ Detached $\rangle$ gives

1.1 $\langle \varepsilon \mid s_0\text{: Detached}_A\rangle$

1.2 $\langle \varepsilon \mid s_0\text{: Detached}_B\rangle$

1.3 $\langle \varepsilon \mid s_0\text{: Detached}_C\rangle$

2. Applying KB to rule $\text{Detached}_A \sqsubseteq \exists\text{Attach}_A.\text{Attached}_A \sqcap \text{Idle}_A$ produces

$\langle \varepsilon \mid s_0\text{: } \neg\text{Detached}_A \sqcup (\exists\text{Attach}_A.\text{Attached}_A \sqcap \text{Idle}_A)\rangle$.

Applying OR gives two branches:

2.1 $\langle 0 \mid s_0\text{: } \neg\text{Detached}_A\rangle$ which is closed.

2.2 $\langle 1 \mid s_0\text{: } \exists\text{Attach}_A.\text{Attached}_A \sqcap \text{Idle}_A\rangle$. Applying AND gives

2.2.1 $\langle 1 \mid s_0\text{: Idle}_A\rangle$

2.2.2 $\langle 1 \mid s_0\text{: } \exists\text{Attach}_A.\text{Attached}_A\rangle$ to which applying SOME produces $\langle 1 \mid s_0 \text{ Attach}_A \text{ } s_1\text{: Attached}_A\rangle$

3. We derive $\langle 1 \mid s_0\text{: } \neg(\text{Attached}_A \sqcap \text{Idle}_A) \sqcup \text{T\_Null}\rangle$ in an intermediate step, by applying KB. Then applying OR:

3.1 $\langle 10 \mid s_0\text{: } \neg(\text{Attached}_A \sqcap \text{Idle}_A)\rangle$ (closed as to 2.2).

3.2 $\langle 11 \mid s_0\text{: T\_Null}\rangle$

4. We derive

$\langle 11 \mid s_0\text{: } \neg\text{T\_Null} \sqcup \exists\text{T\_Att\_Auth}_{AB}.\text{T\_Call\_Handling}_{AB}\rangle$ in an intermediate step, by applying KB. Then applying OR:

4.1 $\langle 110 \mid s_0\text{: } \neg\text{T\_Null}\rangle$ (closed).

4.2 $\langle 111 \mid s_0\text{: } \exists\text{T\_Att\_Auth}_{AB}.\text{T\_Call\_Handling}_{AB}\rangle$ to which we apply SOME resulting in

$\langle 111 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ : T\_Call\_Handling}_{AB}\rangle$ with new state s_1.

5. We derive

$\neg\text{Detached}_C \sqcup \forall\text{T\_Call\_Handling}_{AB}.\text{Detached}_C$ in an intermediate step by applying KB to the frame axioms. Then applying OR:

5.1 $\langle 1110 \mid s_0 \text{ } \neg\text{Detached}_C\rangle$ (closed).

5.2 $\langle 1111 \mid s_0 \text{ } \forall\text{T\_Call\_Handling}_{AB}.\text{Detached}_C\rangle$ to which we apply rule ALL, resulting in

5.3 $\langle 1111 \mid s_0 \text{ T\_Call\_Handling}_{AB} \text{ } s_1 \text{: Detached}_C\rangle$.

6. We need further intermediate derivation. Applying $\text{T\_Call\_Handling}_{AB} \sqcap \text{Detached}_C \sqsubseteq \exists\text{T\_Called\_Party\_Busy}_{AC}.\text{T\_Exception} \sqcap \neg\text{Notified}_A$ of the (MCA-modified) TBox gives

$\langle 1111 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ : } \neg(\text{T\_Call\_Handling}_{AB} \sqcap \text{Detached}_C) \sqcup \exists\text{T\_Called\_Party\_Busy}_{AC}.\text{T\_Exception} \sqcap \neg\text{Notified}_A\rangle$. Applying OR:

6.1 $\langle 11110 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ : } \neg(\text{T\_Call\_Handling} \sqcap \text{Detached}_C)\rangle$ (closed).

6.2 $\langle 11111 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ : } \exists\text{T\_Called\_Party\_Busy}_{AC}.\text{T\_Exception} \sqcap \neg\text{Notified}_A\rangle$ to which applying AND results in

6.2.1 $\langle 11111 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ : } \exists\text{T\_Called\_Party\_Busy}_{AC}.\text{T\_Exception}\rangle$ to which we apply SOME resulting in $\langle 11111 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ T\_Called\_Party\_Busy}_{AC} \text{ } s_2\text{: T\_Exception}\rangle$

6.2.2 $\langle 11111 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ : } \neg\text{Notified}_A\rangle$

7. We derive $\text{T\_Exception} \sqsubseteq \text{T\_Null}$ to which applying KB and then OR results in

$\langle 111111 \mid s_0 \text{ T\_Att\_Auth}_{AB} \text{ } s_1 \text{ T\_Called\_Party\_Busy}_{AC} \text{ } s_2\text{: T\_Null}\rangle$

8. We derive from (MCA-modified) Tbox the rule

$\neg Notified_A \sqcap Detached_B \sqsubseteq \exists Attach_B.Attached_B \sqcap Idle_B \sqcap \exists Notify_{AB}.Notified_A$ which produces

$\langle 111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2: \neg Notified_A \sqcap Detached_B \sqsubseteq \exists Attach_B.Attached_B \sqcap Idle_B \sqcap \exists Notify_{AB}.Notified_A \rangle$. Applying KB produces

$\langle 111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2: \neg(\neg Notified \sqcap Detached_B) \sqcup \exists Attach_B.Attached_B \sqcap Idle_B\ \exists Notify_{AB}.Notified_A \rangle$. Applying OR:

8.1 $\langle 1111110 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2: Notified_A \sqcup \neg Detached_B \rangle$ (closed).

8.2 $\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2: \exists Attach_B.Attached_B \sqcap Idle_B \sqcap \exists Notify_{AB}.Notified_A \rangle$ to which we apply AND resulting in

8.2.1 $\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2 : \exists Attach_B.Attached_B \rangle$

to which applying SOME results in

$\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Attach_B\ s_3: Attached_B \rangle$.

8.2.2 $\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2 : Idle_B \rangle$

8.2.3 $\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2 : \exists Notify_{AB}.Notified_A \rangle$

to which applying SOME results in

$\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3: Notified_A \rangle$.

9. We derive $T\_Null \sqsubseteq \exists T\_Att\_Auth_{AB}.T\_Call\_Handling_{AB}$ and apply KB which results in

$\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3: \neg(T\_Null \sqcap Idle_B) \sqcup \exists T\_Att\_Auth_{AB}.T\_Call\_Handling_{AB} \rangle$. Apply OR:

9.1 $\langle 11111110 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3: \neg(T\_Null \sqcap Idle_B) \rangle$ (closed).

9.2 $\langle 11111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3: \exists T\_Att\_Auth_{AB}.T\_Call\_Handling_{AB} \rangle$ to which applying SOME produces $\langle 11111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3\ T\_Att\_Auth_{AB}\ s_4 : T\_Call\_Handling_{AB} \rangle$.

10. We derive $\langle 11111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3\ T\_Att\_Auth_{AB}\ s_4 : \neg(Term\_Call\_Handling_{AB} \sqcap Detached_C) \sqcup \exists T\_Called\_Party\_Busy_{AC}.T\_Exception \rangle$ in an intermediate step by applying KB. Apply OR:

10.1 $\langle 111111110 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3\ T\_Att\_Auth_{AB}\ s_4 : \neg(Term\_Call\_Handling_{AB} \sqcap Detached_C) \rangle$ (closed).

10.2 $\langle 111111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3\ T\_Att\_Auth_{AB}\ s_4 : \exists T\_Called\_Party\_Busy_{AC}.T\_Exception \rangle$ to which applying SOME results in $\langle 1111111 |\ s_0\ T\_Att\_Auth_{AB}\ s_1\ T\_Called\_Party\_Busy_{AC}\ s_2\ Notify_{AB}\ s_3\ T\_Att\_Auth_{AB}\ s_4\ T\_Called\_Party\_Busy_{AC}\ s_5 : T\_Exception \rangle$.

11. At last we derive $MCA_B \sqsubseteq Notified_A \sqcap Idle_B \sqcap \neg T\_Exception$ and with 8.2.2, 8.2.3 and 10.2 we finish the closure.

The result is a closed tableau which means that $\delta_{CFU}(\delta_{MCA}(MCS))$ interacts on activation $\{MCA_B\} \cup \{CFU_B\}$. It is important to mention that the service interaction can be detected automatically since the programmability of the algorithm.

6. Conclusion

The description of the mobile communication system behavior as processes related to CAMEL basic call processing and mobility management allows interaction with the CAMEL service logic. The suggested approach to representation of basic states of a party involved in a call and attach-detach states of a mobile terminal provides means for detection of interactions between call-related and call unrelated CAMEL services.

CAMEL also offers possibility of triggering services as a result of messaging events and resource reservation for data communication (PDP context setup). The SMS state model makes it possible to start CAMEL services as the result of sending a short message. The PDP context state model allows CAMEL service to control the process of resource reservation for data communications.

Following the suggested approach both SMS state model and PDP context model may be described as logical rules. Representing the CAMEL state models by description logic provides formal description of mobile communication system's behavior.

Service behavior may be described by refinements and modifications. By applying standard inference algorithms, service interaction can be detected automatically.

Acknowledgment

The research is partially supported by the Project 102ni107-7 funded by Research and Development Sector at Technical University of Sofia, Bulgaria.

References

1. Chi, C., R. Hao. Test Generation for Interaction Detection in Feature-rich Communication Systems. - *Computer Networks: International Journal of Computer and Telecommunications Networking*, 51, 2007, 2, 426-438.

2. Chi, C. An Optimization Technique for Feature Interaction Test Generation. Feature Interactions in Telecommunications and Software Systems VIII. Edited by Stephan Reiff-Marganiec, Mark D. Ryan, 2005, 298-304.

3. Cavalli, A., S. Maag. A New Algorithm for Service Interaction Detection Formal Methods and Software Engineering. Proc. of the 4th International Conference on Formal Engineering Methods: Formal Methods and Software Engineering, 2495, 2002, 371-382.

4. Leelaprute, P., T. Matsuo, T. Tsuchiya and T. Kikuno. Detecting Feature Interactions in Home Appliance Networks. Proc. of ACIS International Conference Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008, 895-903.

5. Felty, A. Temporal Logic Theorem Proving and its Application to the Feature Interaction Problem.Proc. of the IJCAR Workshop on Issues in the Design and Experimental Evaluation of Systems for Modal and Temporal Logics. Technical Report DII 14/01, University of Siena, 2001, <http://www.site.uottawa.ca/ ~afelty/dist/ijcar01.pdf>.

6. Li, H., S. Krishnamurthy, K. Fisler. Verifying Cross-cutting Features as Open Systems. - *ACM SIGSOFT Software Engineering Notes*, 27, 2002, 6, 89-98.

7. Reiff-Marganiec, S., J. Turner. Feature Interaction in Policies - *Computer Networks,* 45, 2004, 5, 569-584.

8. Calder, M. and A. Miller. Feature Interaction Detection by Pairwise Analysis of LTL Properties - a Case Study. - *Formal Methods in System Design*, 28, 2006, 3, 213-261.

9. Calder, M., M. Kolberg, E. Magill and S. Reiff-Marganiec. Feature Interaction: a Critical Review and Considered Forecast. - *Computer Networks*, 41, 2003, 115-141.

10. Krivoi, S. An Algebraic Approach to Managing Feature Interaction in Telecommunication Systems. Proc. of International Conference on Parallel Computing in Electrical Engineering (PARELEC'02*)*, 2002, 99.

11. Fu, Q., P. Harnois, L. Logrippo and J. Sincennes. Feature Interaction Detection: a LOTOS-based Approach. - *Computer Networks*, 32, 2000, 4, 433-448.

12. Calder, M. and M. Kolberg, and E. Magill, and D. Marples, and S. Reiff-Marganiec. Hybrid Solutions to the Feature Interaction Problem. Feature Interactions in Telecommunications and Software Systems VII, IOS Press, 2003.

13. Lorentsen, L., A. Tuovinen, J. Xu. Modeling of Features and Feature Interactions in Nokia Mobile Phones Using Colored Petri Nets. Application and Theory of Petri Nets, Springer, 2002, 15-45.

14. Lorentsen, L., A. Tuovinen, J. Hu. Modeling Feature Interactions Using in Nokia Mobile Phones. Proc. of ICATPN 2002, Lecture Notes on Computer Science, Springer-Verlag Berlin Heidelberg, 2360, 2002, 294-313.

15. Nepomniaschy, V., D. Beloglazov, T. Churina and M. Mashukov. Using Coloured Petri Nets to Model and Verify Telecommunications Systems. Computer Science - Theory and Applications**,** Springer Berlin Heidelberg, 5010, 2008, 360-371.

16. Lohmann, N. A Feature-Complete Petri Net Semantics for WS-BPEL 2.0. Web Services and Formal Methods, Forth International Workshop, WS-FM 2007 Brisbane, Australia, 2007, Lecture Notes in Computer Science, Springer-Verlag, http://www2.informatik.hu-berlin.de/top/ download/publications/Lohmann2007_wsfm.pdf (online).

17. Noldus, R. CAMEL Intelligent Networks for the GSM, GPRS and UMTS Network. John Wiley & Sons Ltd, 2006.

18. Areces, C., W. Bouma, M. de Rijke. Feature Interaction as a Satisfiability Problem.http://www.loria.fr/~areces/ content/papers/ files/fi-mascots.pdf (online).

19. 3GPP TS 22.078 Customized Applications for Mobile Network Enhanced Logic (CAMEL). Phase 4, Stage 2 (Release 10), 2010.

Manuscript received on 09.12.2010

Ivaylo Atanasov *received M.S. degree in electronics from Technical University of Sofia, and PhD degree in communication networks. The current position is Associate Professor at Faculty of Telecommunications. The main research focus is set on open service platforms for next generation networks.*

Contacts:
Technical University of Sofia
tel: +359 2 965 2050
e-mail: iia@tu-sofia.bg

Evelina Pencheva *received M.S. degree in mathematics from University of Sofia „St. Kliment Ohridski", and PhD degree in communication networks from Technical University of Sofia. The current position is Professor at Faculty of Telecommunications. The interests include next generation mobile applications and middleware platforms.*

Contacts:
Technical University of Sofia
tel: +359 2 965 3695
e-mail: enp@tu-sofia.bg