Second Feature Interaction Contest

Contest Committee
Mario Kolberg(1), Evan H. Magill(1) (chair), Dave Marples(2) and Stephan Reiff(3)

(1)University of Strathclyde,
Department of Electronic and Electrical Engineering,
204 George Street, Glasgow G1 1XW, Scotland, UK
{m.kolberg, e.magill}@eee.strath.ac.uk

(2)Telcordia Technologies
445 South Street 1J226B, Morristown, New Jersey 07960
dmarples@research.telcordia.com

(3)University of Glasgow,
Department of Computing Science,
8-17 Lilybank Gardens, Glasgow G12 8RZ, Scotland, UK
sreiff@dcs.gla.ac.uk

Abstract. Following the First Feature Interaction Detection Contest held in 1998, a Second Feature Interaction Contest is held in conjunction with the Sixth International Workshop on Feature Interactions in Telecommunications and Software Systems. The goal is to provide a simple comparison of different automated tools for feature interaction detection. A preferable outcome will be a catalog of interactions. In addition to the usual two-way interactions, participants are encouraged to consider interactions between three features.

This document contains the contest instructions. The Appendixes A, B, and C contain the Basic Call and feature definitions as state diagrams.

1 Introduction

Many feature interaction handling approaches have been proposed during the previous years. Many of these approaches have been implemented into tools. However in order to show the applicability of the approaches, people had to use known interactions. Also because different people use different features to test their tools a comparision of tools is difficult. The goal of this Feature Interaction Contest is to help people to test their tools more widely and also to provide a benchmark of feature interactions which should ease to compare tools from different sources. Also, the contest provides a means to test tools against the original need behind the research - to detect *unknown* interactions. This is because the features are provided by a different source than the tools.

Usually experiments only focus on finding interactions between two features, that is only pairwise interactions are targeted. In order to progress work towards finding interactions between a multiple set of features (more than two), participants are encouraged to also consider interactions between three features (or three instances of features). It is hoped that this will help to show the true potential of a number of tools and the respective approaches.

As with the previous contest, this will be split in two phases. For phase one, ten features are made available on August 23rd, 1999. The results for phase one need to be submitted by January 28th, 2000. For phase two, specifications of two more features are made available on January 31st, 2000. You are asked to find the additional interactions and hand in your results by February 14th, 2000.

2 The Network

The employed model of the network is depicted in figure 1. The central element is the *Message Switching*. All the customer end devices (telephones in our case) are connected to it. Each line has associated *Control Software* which can be split into *Basic Call Software* and *Features*. All these instances of *Control Software* are connected to the *Message Switching*. Finally, there is a *Billing System* connected to the *Message Switching*. As needed for the billing, a global clock provides the current time which is accessed through a variable *time*.

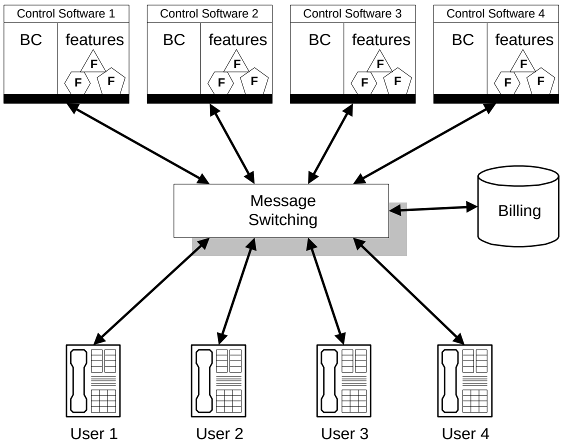

Figure 1: Network Model

All the events and signals exchanged between components of the model are passed through the Message Switching. The following sections describes these signals and their handling in more detail.

3 Events used in the System

The events used in the model can be split into three groups. Firstly, all messages prefixed with an "*o_*" are sent by a Control Software to the Message Switching and the ones prefixed with an "*i_*" are received by a Control Software from the Message Switching. Secondly, events whose names start with "*billing_*" are sent to the Billing Machine. Thirdly, all other messages are used between a user (that is the associated telephone) to the corresponding Control Software. The next three subsections are focussing on the various events.

To obtain a unique message format, all messages consist of four parts: the event, the origin, the destination and an argument. A "-" denotes a constant value for an unneeded argument. The origin and destination might be the same, in which case the message is communicated between the user equipment and the associated Control

Software. The billing messages do not use the origin/destination parameter assignment, details are shown in the following sections.

For the explanations below three variables, X, Y and Z are used. These symbolise three users. However, these variables do not reflect roles of users like originating or terminating party. In other words, X is not necessarily the originating party nor is Y the terminating one.

3.1 User Messages (no Prefix)

A telephone is used as device which provides the interface between the user and the network model. It is important to note that these events are not changed in any way by the Message Switching. All events belonging to this group are listed and explained below, whereby we distinguish between messages initiated by the user and the those initiated by the control software.

User initiated messages:

(offhook, X, X, -) means that subscriber X has gone offhook.

(dial, X, number(Y)) means that subscriber X is dialling the number of subscriber Y.

(onhook, X, X, -) means that subscriber X has gone onhook.

(flash, X, X, -) means that subscriber X flashes, that is goes onhook briefly and then offhook again. Usually this behaviour is created by pressing the flush button on a phone.

Control Software initiated messages:

(dial_tone, X, X, -) means that subscriber X receives a dialtone.

(ringtone, X, X, -) means that subscriber X receives a ringtone.

(busytone, X, X, -) means that subscriber X receives a busytone.

(timeout_tone, A, A, -) means that subscriber X receives a timeout_tone.

(disconnect_tone, X, X, -) means that subscriber X receives a disconnect_tone.

(connect, X, X, -) means that subscriber X's telephone connects to the other party.

(stop_alert, X, X, -) means that subscriber X's telephone stops ringing.

(alert, X, X, -) means that subscriber X's telephone starts ringing.

(announce, X, X, message) means that some announcement is played to X. For instance, X could be notified that the called party is not available but he can leave a message.

(cwtone, X, X, -) means that X receives a Call Waiting tone signalling that the call is currently on hold or that another call attempt is being made (depending on the situation of the user).

(store_msg, X, X, -) means that the message (voice) from the calling party is storred in X's mailbox.

(store_read, X, X, msg) means that the message (voice) from X's mailbox is transmitted to X's telephone.

(store_clear, X, X, -) means that the message stored in X's mailbox is removed.

3.2 Control Software Messages

As said earlier messages commencing with "*o_*" are send by an instance of the control software, those starting with "*i_*" are received. The reader will see that all of these messages come in pairs, which is due to the fact that they are the same message, just with a different direction. The Message Switching converts "*o_*"-messages into "*i_*"-messages.

(o_alert, X, Y, -), (i_alert, X, Y, -) is exchanged between X and Y to notify Y of an call attempt being made. The special case of *(o_alert, X, Y, anonymous)* is used by the calling number delivery blocking and will be converted into *(i_alert, anonymous, Y)* by the message switching to keep the originator private. The message switching keeps track of the association between the anonymous and the return address needed for further message exchange. That is, also the billing machine is always sent the real identity of users and not the *anonymous*.

(o_stopalert, X, Y, -), (i_stopalert, X, Y, -) shows that X is no longer trying to connect to Y

(o_disconnect, X, Y, -), (i_disconnect, X, Y, -) tells the other party about the termination of a call (that was connected)

(o_connect, X, Y, -), (i_connect, X, Y, -) tells the originating party that the called party answers the call

(o_timeout, X, Y, -), (i_timeout, X, Y, -) tells the originating party that the called party does not answer, and that the system timed out the connection attempt.

(o_busy, X, Y, -), (i_busy, X, Y, -) informs about the busy status, in this case the called party is busy.

(o_free, X, Y, -), (i_free, X, Y, -) informs about the busy status, in this case the called party is free.

(o_inform, X, Y, information), (i_inform, X, Y, information) is used to communicate messages. It is used here for features that expect an announcement or other notification to be made at the other side of the call.

(o_msg, X, Y, msg), (i_msg, X, Y, msg) is used for the voicemail feature to transmit messages to be stored.

(o_notify, X, Y, Z), (i_notify, X, Y, Z) is used to inform Y that his call with X has been forwarded to Z.

3.3 Billing Messages

Billing events are sent from the Control Software of one user. In the Message Switching these events are directed to the Billing Machine. Typically, billing related events carry three parameters. The first two specify two users who are involved in some activity for which a charge is created. The global time is passed in as third parameter so that the duration of some activity can be measured and a corresponding charge levied.

(billing_start, X, Y, time) means that user X is started to be charged for a call to user Y. However, these roles can be changed by other events (e.g. billing_reverse which have been sent before). The global time is passed in as a parameter so that the billing system can work out the duration of the call upon reception of the billing_stop event.

(billing_stop, X, Y, time) is the counterpart to the billing_start event. Upon reception of this event the duration of a call can be calculated and user X be charged with the correct amount. Please note that this fixed role of X being charged can be changed by other events which are always be sent before the billing_start event.

(billing_forwarded, X, Y, Z) indicates to the billing system that a call from user X to user Y has been forwarded to user Z. Hence the forwarding can be taken into account for billing.

(billing_reverse, X, Y) notifies the billing system that for the next event (billing_start, X, Y, time) not user X but user Y is charged. Also a surcharge might be added to user Y's bill.

(billing_split, X, Y, factor) notifies the billing system that for the next event (billing_start, X, Y, time) users X and Y are charged according to the factor passed in as third parameter. For instance, user Y will pay 30% of the charge if the factor is 30. In addition the *factor* can be used to code policies like the caller only pays a local call charge and the subscriber the rest etc.

(billing_onhook, X, X, time) means that the billing system gets notified of user X going onhook.

(billing_offhook, X, X, time) means that the billing system gets notified of user X going offhook.

In case the billing system does not receive a *billing_start/billing_end* pair after any of the signals *(billing_forwarded, X, Y, Z)*, *(billing_reverse, X, Y)* or *(billing_split, X, Y, factor)*, and before the next *billing_onhook* the concerned call never took place. In this case the block of billing instructions starting from the *billing_offhook* until the *billing_onhook* is erased from the database in the Billing Machine.

3.4 Some Simplifying Assumptions

For the sake of simplicity of both the model and the instructions some assumptions on the functioning of the overall system have been made. These are listed below.

1. The above messages are all there are.

2. There are no network busy conditions.
3. There is no provisioning or de-provisioning of features. Also, there is no feature activation or de-activation. A subscriber either has a feature or has not.
4. The end-user equipment has a button for "Flash".
5. Transitions between states take no time.
6. All transistions without triggering events do not consume time and happen without time delay. That is, no external events can occur in these states and hence such events are not handled.
7. Tones are stopped with the next triggering event.

4 Notation used for Defining POTS and Features

The Basic Call Model and all features are defined as a set of events on interfaces between network components (telephones, Control Software, Message Switching and Billing Machine). The definitions are expressed in the form of state transition diagrams. Nodes in these diagrams represent states. State transitions are triggered by an incoming event. During the transition some events may be sent to various network elements. The trigger event is typed in *italics* in the diagrams. Outgoing events are typed in normal font. Event sequences involving multiple calls can be interleaved to define global system activity.

As a base a basic two-party POTS diagram is given (cf. appendix A). The diagram consists of two half call models, one representing the originating and the other the terminating side of the call. Beside their names, all states have a unique number assigned (BC 1 - BC 12). These numbers are referred to in the features in order to link to the Basic Call Model. All state names prefixed with an "*o_*" belong to the originating half of the call model whereas the states prefixed with "*t_*" belong to the terminating half. The state *idle* is shared between both half models.

All nodes are connected by directed edges. An edge represents a transition from one state two the next. Clearly, a transition which corresponds to an incoming event can be performed. In addition, in some states transitions are triggered by state internal triggers. An example is the *o_timeout* edge from BC 10 to BC 1, where some kind of timer is assumed that triggers the transition after a certain time has elapsed.

Each feature modifies POTS somehow. As stated previously we do not model service provisioning or activation, so if a subscriber has a feature, that feature is always active. Feature diagrams originate and terminate at some Basic Call node. That is, the behaviour of a feature extends the behaviour of the Basic Call Model. The link from the Basic Call Model to a feature is done by introducing a new transition from a Basic Call state or by *replacing* one or more transitions. That is for the latter, only the transition which is used to link the BCM and the feature is replaced. All other transitions which might be possible from the BCM state are still valid. For instance in the feature *Ringback when Free* the transition triggered by the *(onhook, A, A, -)* event from state BC 5 to BC 1 is *replaced* by a transition to the state RB 1 (check RBlist). In other words the original transition from the Basic Call is *not valid* with that feature. On the other hand, all other transitions possible from state BC 5 are still valid with this feature. The (i_inform, B, A, "ringback") transition from BC 5 to BC 8 introduced in the Ringback when Free feature is additional to the transitions defined in the BCM for state BC 5.

5 An Outline of the Phase 1 Features

5.1 Call Forwarding on Busy

With this feature all calls to the subscriber's line are redirected to a predetermined number when the subscriber's line is busy. The Billing Machine is notified of the forwarding and it is assumed that the subscriber pays the charge for the forwarded call from his location to the location where the call has been forwarded to (this assumption is part of the Billing Machine and as such the implementation is not considered here). The definition can be found in appendix B.1.

5.2 Teenline

During a pre-set time of day, this feature restricts all outgoing calls from the subscriber's telephone. To place an outgoing call during that time a PIN is required. The definition of the feature can be found in appendix B.2.

5.3 Terminating Call Screening

By this feature the originators of all incoming calls to the subscriber's telephone are screened against a screening list. If an originator of an incoming call matches an entry in the list an announcement is played to that line and the call is cancelled. The definition of the feature can be found in appendix B.3.

5.4 Call Waiting

This feature allows the subscriber to be notified of an incoming call while he is busy in a conversation and to accept the new call be putting the originating call on hold. The subscriber is then able to toggle between these two calls (cf. appendix B.4).

5.5 Three Way Calling

Three way calling allows a user already connected to another user to bring a third partner into the call. Any side of the first call (as long as its the subscriber to 3WC) can setup a connection to the new party, by putting the current partner on hold, connecting to the third side and then joining both lines. The three way call is terminated with any party going onhook. The feature can be found in appendix B.5.

5.6 Reverse Charging

Reverse charging is also known as freephone billing, and allows the subscriber to be charged for all calls in which the subscriber is the terminating party. The definition of the feature can be found in appendix B.6.

5.7 Calling Number Delivery Blocking

Usually the callers identity is available at the terminating side, for evaluation by the callee if required (e.g. caller number display or terminating call screening). This feature blocks the provision of the callers number at the terminating side. The definition of the feature can be found in appendix B.7.

5.8 Ringback when Free

When a call attempt is made to a busy line with this feature subscribed, the caller is informed that he will be called back, as soon as the other person becomes available. Once the subscriber terminates his/her current call a connection to the stored numbers will be established. The definition of the feature can be found in appendix B.8.

5.9 Voice Mail

Voice mail works similar to an answering machine, by offering the possibility to leave messages if the called user is not answering. The stored messages can be listened to by the subscriber. The definition of the feature can be found in appendix B.9.

5.10 Split Billing

Split billing allows to share costs between the partners in a call. A company might provide local call charge lines to customers as a service, in which case the customer (and originator of a call) pays the local charges and the company the rest. The definition of the feature can be found in appendix B.10.

6 An Outline of the Phase 2 Features

6.1 Call Transfer

Call Transfer allows the subscriber to transfer the current call to a 3rd Party. That is, while being in a call the subscriber can put the second party on hold and can setup a call to a 3rd Party. Once the subscriber goes onhook the second party of the first call and the 3rd Party are connected. This is effectively a mid-call call transfer. The feature supports both, the subscriber being the caller or callee in the first call. The definition of the feature can be found in appendix C.1.

6.2 Group Ringing

This feature allows for an incoming call to ring at three phones. The phone which goes offhook first is connected to the calling party. The remaining two phones stop ringing. The definition of the feature can be found in appendix C.2.

7 Submission of Results

This section summarises the deliverables expected from the contest participants. We would like to support participants in focussing on the real work and hence provide some templates which should be filled in by participants. We aimed at keeping the amount of writing to a minimum while trying to give participants of the workshop maximum gain of the contest results.

7.1 Submissions

At the end of each, Phase 1 and Phase 2, participants are supposed to submit a summary of their results in form of tables, as outlined in section 7.2. Please identify in the tables where interactions occurred, who subscribes to the feature (terminator or originator) and also which states the respective features have been in. The latter is simplified by a consistent numbering of states across the features, and should help us retrace the interaction.

For three way interactions we are only interested in **true** three way interactions, meaning those where the interaction does no longer exist once one feature is removed. In other words, the *same* interaction cannot be occur between any pair of the involved features. For both phases a two to three page document should accompany the table, describing your method, tool and optionally selected interaction scenarios in more detail.

Further, in order to stimulate a lively discussions between workshop delegates, participants are requested to create a poster for exhibition on the conference, giving an overview of the approach, tool and results. For details on posters please refer to the FIW'00 calls for papers and participation.

The participation in the contest does not restrict the contestants in any way to submit papers to the workshop itself. This includes papers on their approach and tool used. This is not part of the contest and will have no impact on the valuation of the results. Please refer to the guidelines given in the call for papers for FIW.

Summarising, there is one table to be filled in (in case you do both three way and two way interactions, there are two), three pages to write and a poster to create. This keeps the submissions to a minimum and allows the participants to spent their time on the more important issue of detecting interactions.

7.2 Table Outlines for Concise Submission of Results

Suggestions for the tabular representation of the results are given for both two-way interactions (Table reftab2way) and three-way interactions (Table 7.2). The table outline for three-way interactions is not complete and is intended to provide only the general outline.

	CFB	CW	3WC	TL	TCS	RC	CNDB	RBF	VM	SB
CFB										
CW										
3WC										
TL										
TCS										
RC										
CNDB										
RBF										
VM										
SB										

Table 1: Suggested Table for Two-Way Interactions

		CFB	CW	3WC	TL	TCS	RC	CNDB	RBF	VM	SB
	CFB										
	CW										
	3WC										
	TL										
CFB	TCS										
	RC										
	CNDB										
	RBF										
	VM										
	SB										
		CFB	CW	3WC	TL	TCS	RC	CNDB	RBF	VM	SB
	CFB										
	CW										
	3WC										
	TL										
CW	TCS										
	RC										
	CNDB										
	RBF										
	VM										
	SB										
...	...										
		CFB	CW	3WC	TL	TCS	RC	CNDB	RBF	VM	SB
	CFB										
	CW										
	3WC										
	TL										
VM	TCS										
	RC										
	CNDB										
	RBF										
	VM										
	SB										
		CFB	CW	3WC	TL	TCS	RC	CNDB	RBF	VM	SB
	CFB										
	CW										
	3WC										
	TL										
SB	TCS										
	RC										
	CNDB										
	RBF										
	VM										
	SB										

Table 2: Suggested Table for Three-Way Interactions

7.3 Reminder of Deadlines

There are three deadlines for submission for contest participants; the results of phase one should be submitted by 28th January 2000 and the results of phase two by 14th February 2000. Posters are due on 17th April 2000. For details on these deadlines please refer to Section 7.1.

A Basic Call Definition

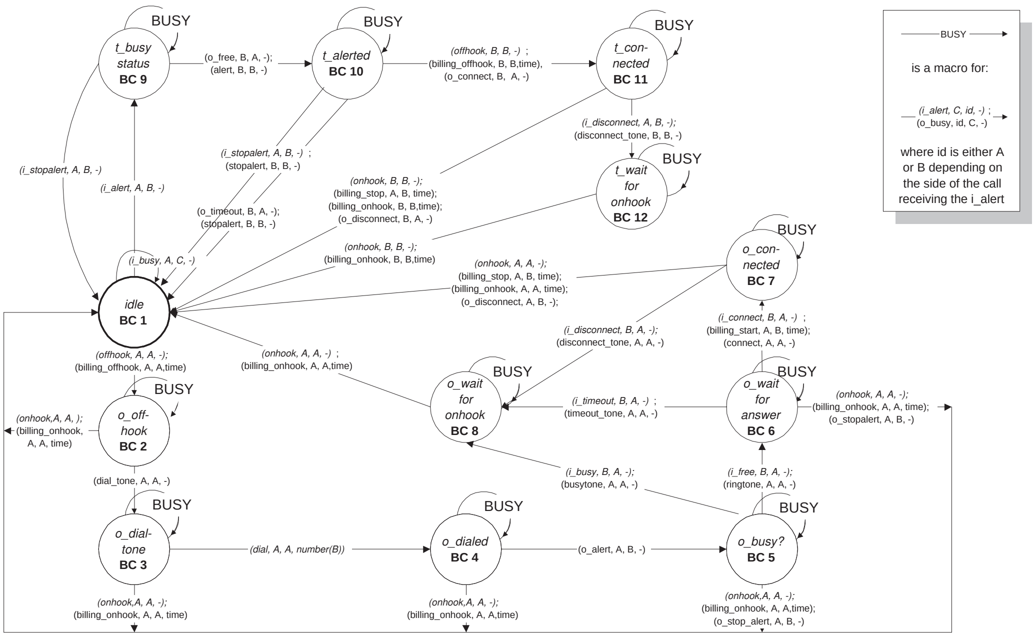

Figure 2: Basic Call Model

B Phase One Feature Definitions

B.1 Call Forwarding on Busy

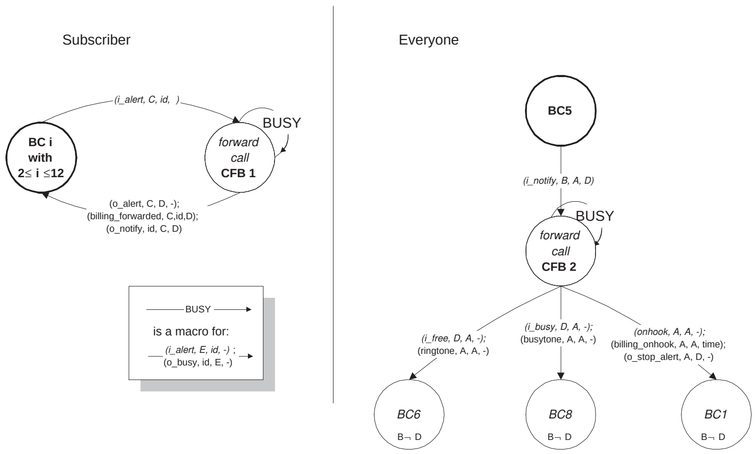

Figure 3: Call Forwarding on Busy Model

B.2 Teenline

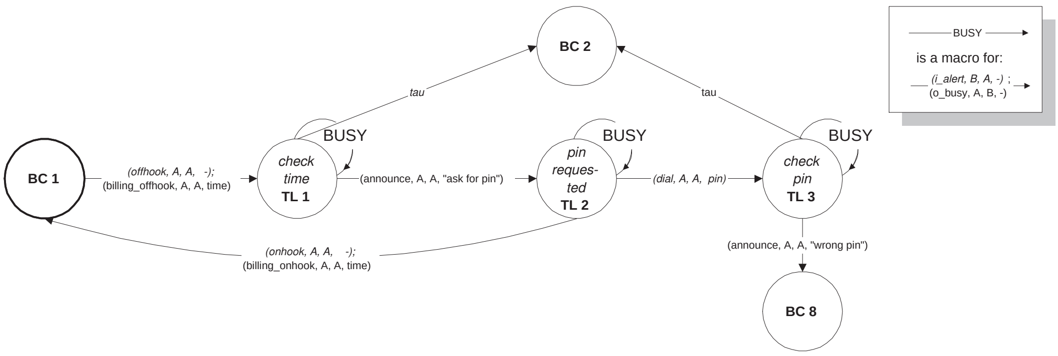

Figure 4: Teenline Model

B.3 Terminating Call Screening

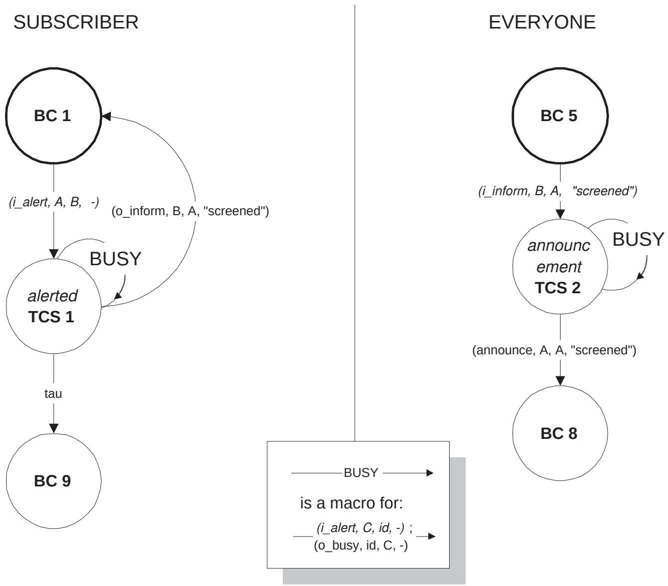

Figure 5: Terminating Call Screening Model

B.4 Call Waiting

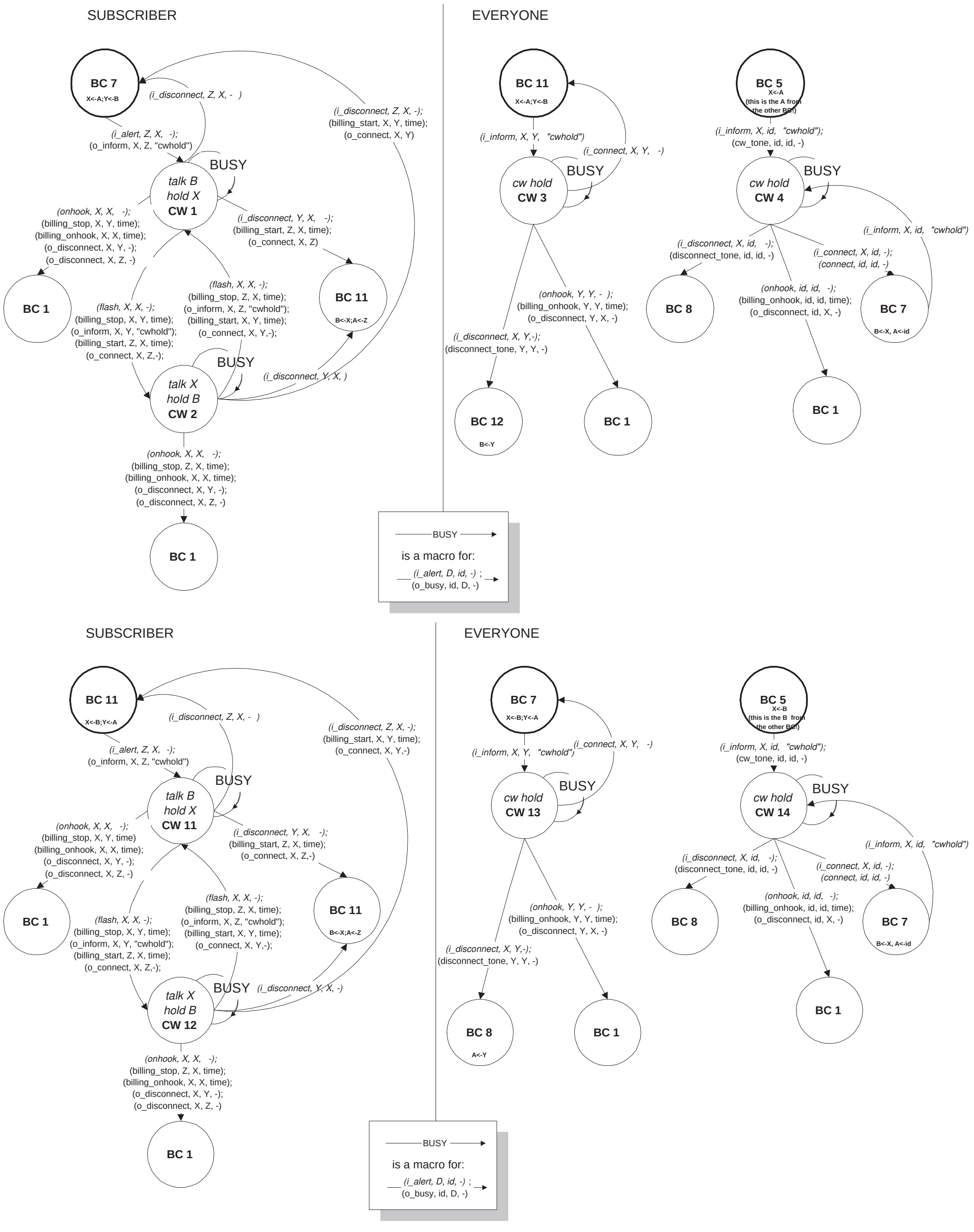

Figure 6: Call Waiting Model

B.5 Three Way Calling

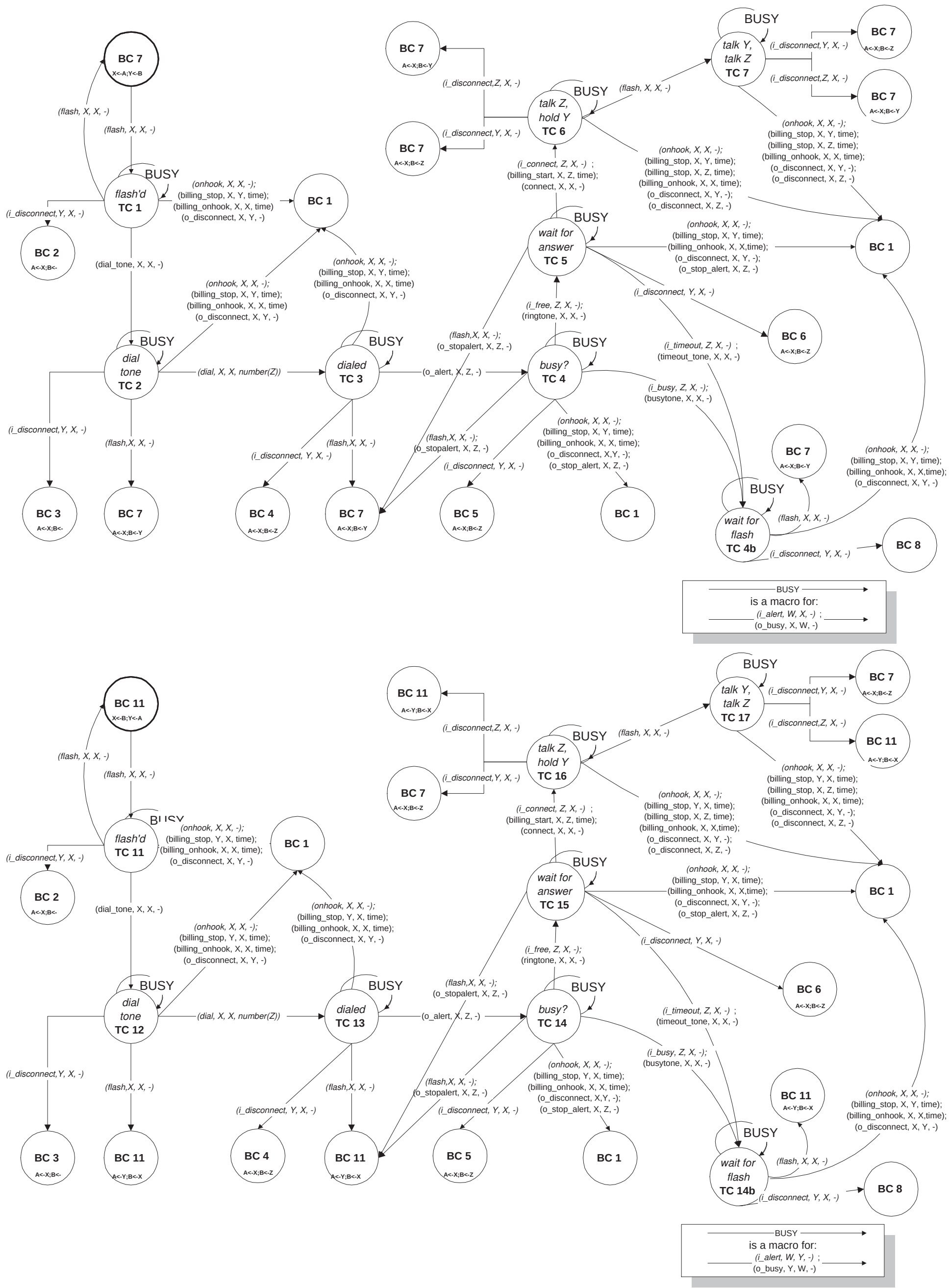

Figure 7: Three Way Calling Model

B.6 Reverse Charging

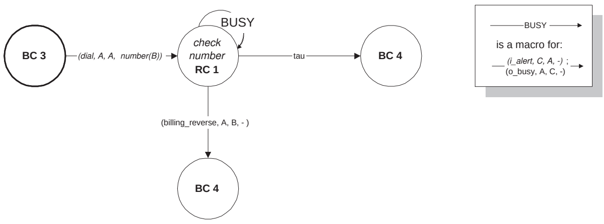

Figure 8: Reverse Charging Model

B.7 Calling Number Delivery Blocking

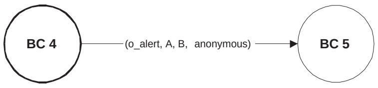

Figure 9: Calling Number Delivery Blocking Model

B.8 Ringback when Free

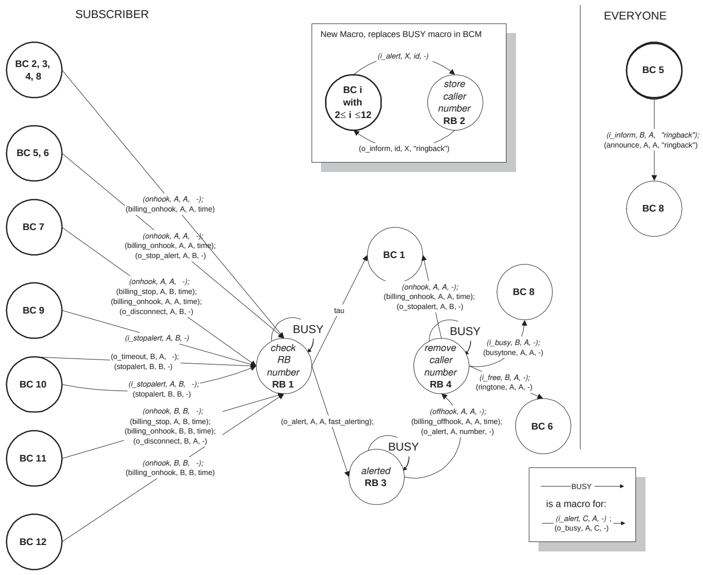

Figure 10: Ringback when Free Model

B.9 Voice Mail

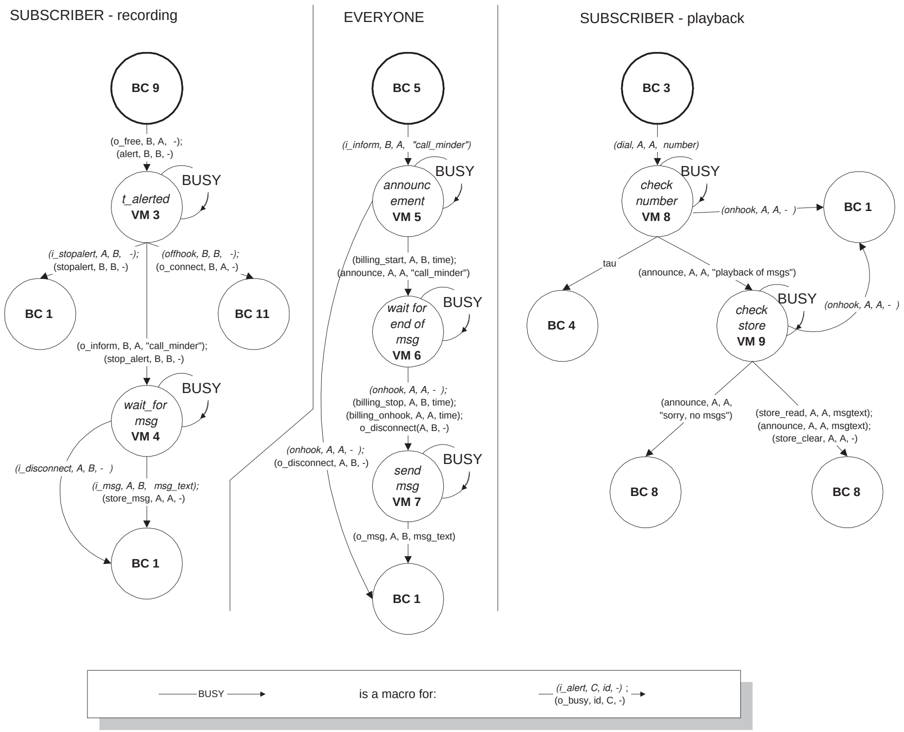

Figure 11: Voice Mail Model

B.10 Split Billing

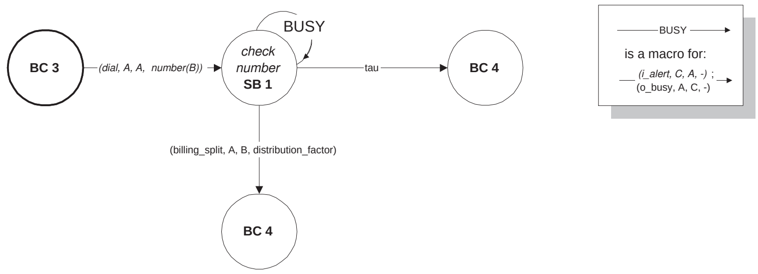

Figure 12: Split Billing Model

C Phase Two Feature Definitions

C.1 Call Transfer

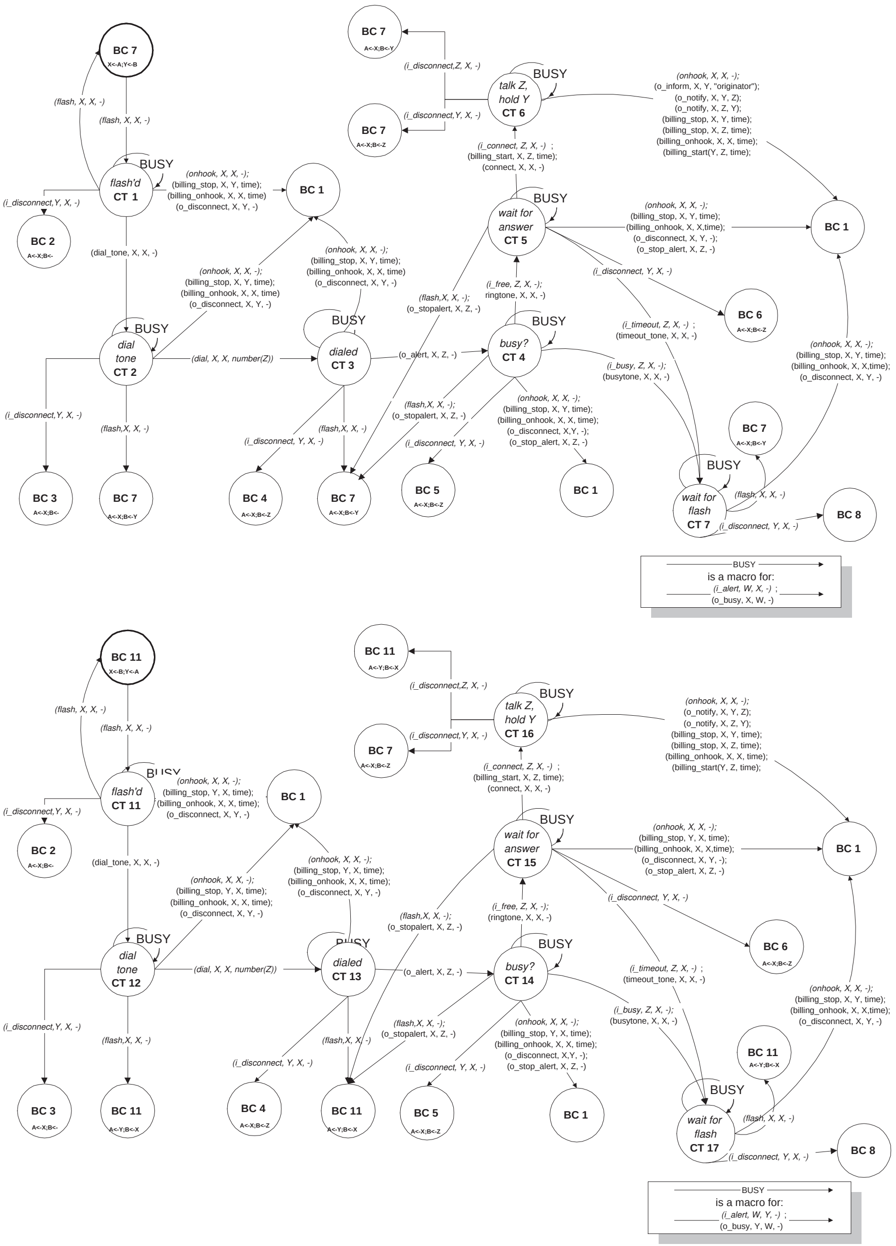

Figure 13: Call Transfer Model - Subscriber

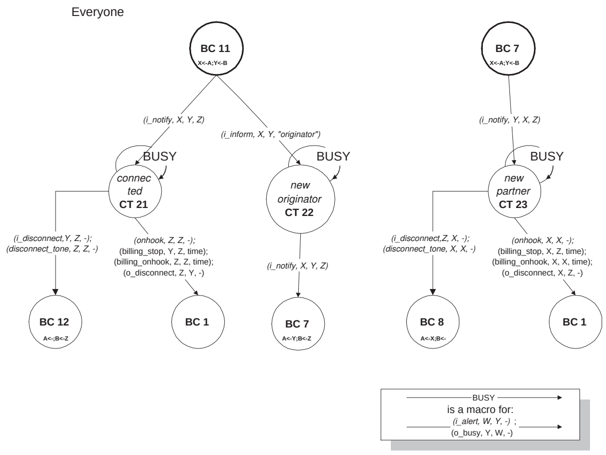

Figure 14: Call Transfer Model - Everyone

C.2 Group Ringing

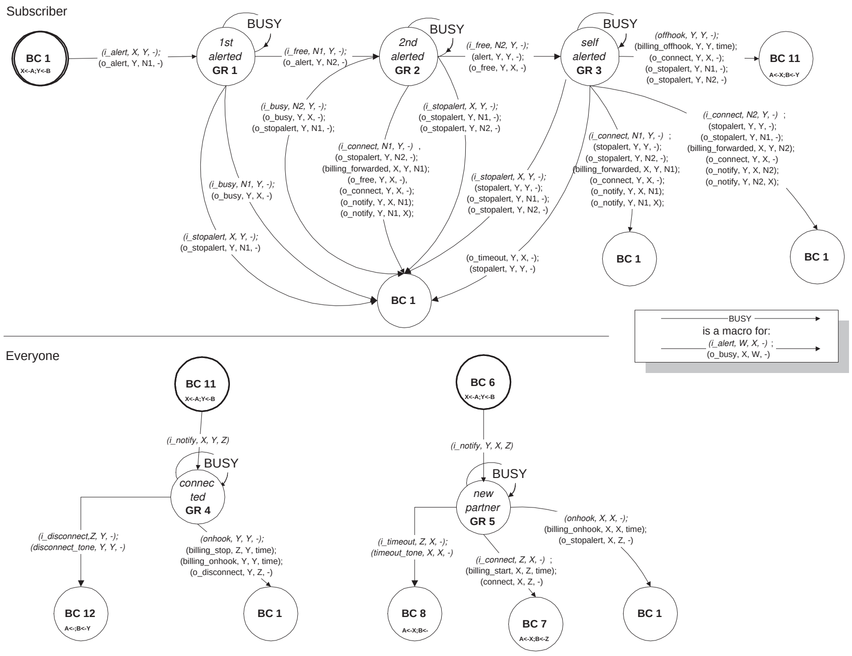

Figure 15: Group Ringing Model