

# Entry for FIW'00 Feature Interaction Contest

Malte Plath and Mark Ryan  
School of Computer Science  
University of Birmingham  
Birmingham B15 2TT  
UK

<http://www.cs.bham.ac.uk/{~mcp,~mdr}>

March 3, 2000

## Abstract

This paper describes the results of our work on the Feature Interaction Contest<sup>1</sup>. It contains the results for both phases. The results of Phase 1 are updated with respect to our prior submission.

## 1 Introduction

The tools we used are FDR<sup>2</sup> and ProBe<sup>3</sup>, developed by Formal Systems Ltd. We also used the Python programming language. FDR and ProBe are tools for model-checking and animating specifications written in  $CSP_M$ , a machine-readable version of Hoare's CSP.

We described the Basic Call Model (as given in the Contest Instructions) in  $CSP_M$ . Our call model is parameterised by a finite set of subscribers. We found we could verify deadlock freedom of the basic call model with four subscribers comfortably within minutes. The time memory required to check the model grows rapidly with the number of users. However, we believe that most of the interactions can be witnessed with three or four subscribers. The effect of adding features, on the other hand, proved rather moderate.

We developed an extension of  $CSP_M$  for describing features, as additions or replacements of transitions in the Basic Call Model, and wrote a script in Python to apply a feature to the Basic Call, resulting in a new version of the Basic Call which incorporates the feature. The script also takes arguments giving the name of the subscriber, and other parameters (such as the forwarding target for CFB, etc). The script can be used to apply a feature to an already-featured system, giving systems with several features. In general, the order of integrating features is significant.

Some effort was spent debugging the basic model, and ensuring that each feature could be integrated individually without producing deadlocks.

To find feature interactions among pairs of features, we used the script to obtain the  $CSP_M$  model for the basic call model with both features integrated. We then used FDR to

---

<sup>1</sup>Organised by the Sixth International Workshop on Feature Interactions in Telecommunications and Software Systems Glasgow, Scotland, United Kingdom, 17–19th May 2000. <http://www.comms.eee.strath.ac.uk/~fi/fiw00/>

<sup>2</sup><http://www.formal.demon.co.uk/FDR2.html>

<sup>3</sup><http://www.formal.demon.co.uk/probe.html>

detect deadlocks. A deadlock represents a potential interaction, which has to be investigated. We could also use ProBe to simulate scenarios, and use FDR and ProBe in conjunction with observer processes to restrict attention to certain types of paths. Our list of pairwise interactions is presented in the next session.

We also attempted to find feature interactions among triples of features, which were not interactions between pairs. We set about this as follows: we looked for triples of features which were pairwise interaction free (taking into account constraints on subscriber identity). This resulted in approximately 10 triples. We eliminated those which contained features which our experience of the pairwise case had shown to be relatively benign. We ended up with just one triple of features: CW/TCS/VM (each with a different subscriber). We were not able to find any interaction (given that constraint on the subscribers). Our conclusion is that the given set of features exhibits so many features in the pairwise case that there are nearly none to look for in the triple case.

## 2 Feature Interactions Phase 1

We present an overview of the feature interactions we found in the following table. The numbers in the table 1 refer to the list of descriptions below. We have grouped some interactions together, most notably the incompatibility between Call Forwarding on Busy, Call Waiting and Ring Back (number 14) where the cause of the interactions was very similar.<sup>4</sup> Please note that we have not listed twice interactions of features which don't commute, such as Split Billing and Reverse Charging.

In this section, we have marked changes with respect to our previous submission by (\*).

8

Table 1: Feature Interactions Phase 1

1. **CNDB<sup>A</sup> & CFB<sup>B</sup>**: When a call from A is forwarded, CNDB is lost, since the outgoing message would have to be (o\_alert,anonymous,D,-). (The contest specification is too imprecise about the translation and necessary book-keeping done by the message switching software. (*cf.* CNDB & RBWF, next item))

NB: In our model we have resolved this interaction: B cannot send an o\_alert message with sender field A or 'anonymous'. We therefore send only the notification back to A, and A moves back to state o\_dialled, where CNDB takes effect. We found the interaction when we considered what difference our solution would make.

2. **CNDB<sup>A</sup> & RBWF<sup>B</sup>**: When A calls B while B is busy, A will get the busy signal – no ringback is set up. (Again, the specification of how anonymous alert messages are handled is too imprecise to be entirely sure what would happen. However, it can be seen as an interaction either way, whether a ringback is registered or not.)

---

<sup>4</sup>Note that this sometimes leads to overlaps between different 'generic' feature interactions, e.g. 18 and 35 obviously describe the same interaction for the feature combination RBWF & TWC.

3. **CNDB<sup>A</sup> & RBWF<sup>A</sup>**: When RBWF sets up a ringback call to B, B will receive A's number.
4. **CNDB & TCS**: When B screens A, A can still call B using CNDB. (Alternatively, TCS could block any calls with CLI - violating the dual property: Callers not on the screening list can get through.)
5. **CNDB<sup>A</sup> & TWC<sup>A</sup>**: CNDB will not work for the second call that is set up, i.e. the third party in the call will receive A's CLI.
6. **CNDB<sup>A,B</sup> & CW<sup>C</sup>**: When C receives two anonymous calls (via Call Waiting), the recipient of outgoing messages cannot be determined. (Message Switching and the communication protocol would have to be far more complex.)  
 Again this is an interaction which our implementation resolves by design, but which became apparent in considering the effects of our design choices.
7. **RC & SB**: are mutually exclusive, regardless of subscriber. Whichever is integrated last overrides the previous one. (They both change the (dial, A, A, number(B)) transition from BC3 for everyone.) We assume, however, that Split Billing would be implemented in such a way that different splitting factors may be set for different subscribers, so that SB will not interfere with itself (for different subscribers).
8. **RC/SB & VM<sup>(\*)</sup>** are mutually exclusive, regardless for the Voice Mail subscriber. Whichever is integrated last overrides the previous one. This means that a VM subscriber either cannot retrieve his/her Voice Mail, or cannot take advantage of Split Billing and Reverse Charging. (Both feature change the (dial, A, A, number(B)) transition from BC3 for the VM subscriber.)
9. **RC/SB & CFB**: forwarded calls (i.e. the second leg) will not benefit from RC or SB. (However, when an RC/SB subscriber forwards, the first call leg is charged correctly to the RC/SB subscriber and the caller.)
10. **RC/SB & RBWF**: the returned call will not benefit from RC or SB, i.e. it will be billed at full price to the RBWF subscriber.
11. **RC/SB & TWC**: the second call will not benefit from RC or SB.
12. **TL<sup>A</sup> & RBWF<sup>A</sup>**: Ringback calls are not subject to TeenLine filtering, i.e. the time is not checked and no pin needs to be entered.
13. **TL<sup>A</sup> & TWC<sup>A</sup>**: The second call is not subject to TeenLine filtering. (*cf.* TL & RBWF)
14. **CFB<sup>A</sup>, CW<sup>A</sup>, RBWF<sup>A</sup>**: are mutually exclusive, since they are all triggered by the message (i.alert, B, A, -) in states BC2-BC12. Whichever is integrated last overrides the previous one(s).
15. **CFB generally**: <sup>(\*)</sup> CFB is only triggered in Basic Call states, so in all (busy) feature states a caller will get a busy signal rather than being forwarded. This holds not only for additional features that the CFB subscriber subscribes to, but also to all features which modify all phones (e.g. Split Billing).

16. **CFB<sup>A</sup> & CFB<sup>B</sup>**: A loop can occur if A forwards to B and B to A. (Longer loops are, of course, possible too).
17. **CW<sup>A</sup> & TCS<sup>B</sup>**: If B calls A while A is connected to someone else, B's call will be put on hold, not screened.
18. **RBWF generally:<sup>(\*)</sup>** A "RingBack" is only triggered in Basic Call states, so in all (busy) feature states a caller will get a busy signal rather than the "ringback" message. This holds not only for additional features that the RBWF subscriber subscribes to, but also to all features which modify all phones (e.g. Split Billing).
19. **RBWF<sup>B</sup> & CFB<sup>A</sup>**: If B calls A while A is busy, and A forwards to B, then B will queue his own call for a ringback. (In our implementation, `i_alert`-messages are only accepted from other phones. Hence this interaction shows up as deadlock.)
20. **RBWF<sup>B</sup> & CFB<sup>A</sup>**: If B's RBWF attempts a ringback to A while A is busy, a deadlock occurs because RBWF cannot handle the `i_notify` message.
21. **RBWF<sup>A</sup> & CW<sup>B</sup>**: If a returned call is handled by Call Waiting on the other side, a deadlock result (as above).
22. **RBWF<sup>A</sup> & RBWF<sup>B</sup>**: A returned call (A to B) may be intercepted by B's RBWF, resulting in a deadlock in RB4 at A (`i_inform`-message cannot be accepted).
23. **RBWF<sup>A</sup> & TCS<sup>A</sup>**: RBWF will set up calls to numbers on the subscriber's screening list, since TCS only intercepts call when A is idle. (Not strictly a violation of the TCS requirement, but probably not what the subscriber would want.)
24. **RBWF<sup>A</sup> & TCS<sup>B</sup>**: RBWF will attempt to return calls to B. If A is on B's screening list, this results in a deadlock, since feature state RB4 cannot accept (`i_inform`, `x`, `y`, "screened").
25. **RBWF<sup>A</sup> & VM<sup>B</sup>**: A returned call may end up at the Voice Mail prompt on the other side, resulting in a deadlock in RB4 (`i_inform`-message cannot be accepted).
26. **TCS<sup>A</sup> & VM<sup>A</sup>**: This is a desirable interaction(!): TCS will prevent callers on A's screening list from leaving Voice Mail messages for A.
27. **TWC<sup>A</sup> & CFB<sup>B</sup>**: in state TC4 the "forwarded" message cannot be handled.
28. **TWC<sup>A</sup> & CW<sup>B</sup>**: in TWC feature states the "cwhold" message cannot be handled. Let A be the TWC subscriber calling B, a CW subscriber, and later C another CW subscriber. For the first call this leads to interactions in all states (TC1–TC7, TC11–TC17), when B uses his/her CW feature to put A on hold.
29. **TWC<sup>A</sup> & CW<sup>C</sup>**: This is very similar to the previous one: in TWC feature states the "cwhold" message cannot be handled. For the second call the interactions begin in TC4/TC14, the first state in which C may send a "cwhold" message to A. Again, all further TC states (TC4–TC7, TC14–TC17) have the same problem.
30. **TWC<sup>A</sup> & RBWF<sup>B</sup>**: in state TC4 the "ringback" message cannot be handled. (see below for another interaction)

31. **TWC<sup>A</sup> & RBWF<sup>A</sup>**: RBWF only attempts to complete one call from the queue at a time, i.e. after each normal call termination, it removes one number from the list. Therefore the queue may never get handled if the subscriber ends each call by hanging up from a TWC state, e.g. pressing the ‘flashhook’ button before going onhook. RBWF does not modify the transitions from TC1, so the queue remains unaltered.
32. **TWC<sup>A</sup> & TCS<sup>B</sup>**: in state TC4 the “screened” message cannot be handled.
33. **TWC<sup>A</sup> & VM<sup>B</sup>**: in state TC4 the “callminder” message cannot be handled.
34. **TWC<sup>A</sup> & VM<sup>A</sup>**: the subscriber cannot use a TWC call to retrieve messages from his/her Voice Mail feature.
35. **TWC generally**: Three Way Calling disables CFB, CW, RBWF from the initial “flashhook” to the final “flashhook” or “onhook”. These are features that act on incoming calls (i.alert messages).
36. **CW<sup>A</sup> & CW<sup>B</sup>**: A cannot be put on hold by B while A has a call on hold, be it the one to B or another one. In states CW1 and CW2 the message (i.inform, A, B, “cwhold”) cannot be handled.
37. **VM<sup>A</sup> & CW<sup>A</sup>**: Call Waiting is not active while the subscriber retrieves messages.
38. **VM<sup>A</sup> & CW<sup>B</sup>**: (This is probably a feature, not a bug.) While B is connected to A’s Voice Mail, B’s Call Waiting is disabled. (similar to the previous one)
39. **VM<sup>A</sup> & TL<sup>A</sup>**: TeenLine may prevent the user from accessing the stored Voice Mail, if the user does not have the pin. (We see this as an interaction, since TeenLine is mainly there to stop children from incurring huge phone bills; Voice Mail calls however are free, apart from the rental fee.)
40. **VM<sup>A</sup> & VM<sup>A</sup>**: only one Voice Mail function can be carried out at a time, i.e. while VM is recording or playing back messages, A is busy for all callers.

### 3 Feature Interactions Phase 2

In this section we list the additional feature interactions between the twelve features of Phase 1 and 2 and the Phase 2 features, Call Transfer and Group Ringing. For simplicity we assume that Group Ringing is always subscribed to by user A or B and that the phones in the “group” are C and D, in this order. It is important to note that many interactions of Group Ringing can only really be witnessed with five or more phones, since Group Ringing already ties up three phones, and various other features are concerned with three phones (CFB, CW, TWC, CT). With four phones, Group Ringing only leaves one phone completely independent of GR. In the list of descriptions we have marked these interactions with a dagger (“†”).

41. **CFB<sup>A</sup> & CT<sup>A</sup>**: CFB is disabled in all feature states of CT.
42. **CFB<sup>A</sup> & CT<sup>B</sup>**: CFB is disabled in states CT21 and CT23. Hence, only if A was the callee in the original basic call, will A still benefit from CFB; for the target of the resulting call as well as A, if A was the caller in the original call, CFB will be disabled.

	CT	GR
CFB	41, 42, 43, 15	15, 65
CNDB	44	66
CW	45, 46, 47	67, 68, 69
RBWF	48, 49, 50, 18	18, 70, 71
RC	51	72
SB	51	72
TCS	52, 53	73, 74
TL	54, 55	
TWC	56, 57, 58	75, 76
VM	59, 60	77, 78
CT	61, 62	63, 64
GR	—	79

Table 2: Feature Interactions Phase 2

43. **CFB<sup>A</sup> & CT<sup>B</sup>**: When B establishes a call leg to A using Call Transfer while A is busy, the `i_notify` message cannot be handled by B in CT4/CT14.
44. **CNDB<sup>A</sup> & CT<sup>A</sup>**: If A calls B using CT, B will receive A's number.
45. **CW<sup>A</sup> & CT<sup>A</sup>**: Call Waiting is disabled in all feature states of CT.
46. **CW<sup>A</sup> & CT<sup>B</sup>**: Call Waiting is disabled in states CT21 and CT23. (*Cf.* interaction 42 between CFB and CT.)
47. **CW<sup>A</sup> & CT<sup>B</sup>**: When B establishes a call leg to A using Call Transfer while A is in state BC7 or BC11, the message (`i_inform`, A, B, "cwhold") cannot be handled by B in CT4/CT14. Subsequently, the same problem would occur as the Call Transfer progresses to CT5/CT15 and CT6/CT16.
48. **RBWF<sup>A</sup> & CT<sup>A</sup>**: RBWF is disabled in all feature states of CT.
49. **RBWF<sup>A</sup> & CT<sup>B</sup>**: RBWF is disabled in states CT21 and CT23. (*Cf.* interaction 42 between CFB and CT.)
50. **RBWF<sup>A</sup> & CT<sup>B</sup>**: When B establishes a call leg to A using Call Transfer while A is busy, the message (`i_inform`, A, B, "ringback") cannot be handled by B in CT4/CT14.
51. **RC/SB & CT**: If a call is transferred to a subscriber to RC or SB, the second leg and hence the resulting call will still be billed at the normal rate, not charged to the RC/SB subscriber in full or in part, respectively.
52. **TCS<sup>A</sup> & CT<sup>B</sup>**: B can establish a call to A from someone on A's screening list.
53. **TCS<sup>A</sup> & CT<sup>B</sup>**: If B is on A's screening list: when B establishes a call leg to A using Call Transfer while A is busy, the message (`i_inform`, A, B, "screened") cannot be handled by B in CT4/CT14.
54. **TL<sup>A</sup> & CT<sup>A</sup>**: calls established using CT are not subject to TeenLine filtering.

55. **TL<sup>A</sup> & CT<sup>B</sup>**: If A is the callee in the original call, A can still end up as the originator of the call resulting from a Call Transfer – and consequently paying for the call, whether authorized or not.
56. **TWC<sup>A</sup> & CT<sup>A</sup>**: are mutually exclusive (both trigger on flash-hook in states BC7 and BC11).
57. **TWC<sup>A</sup> & CT<sup>B</sup>**: if B calls A, and B uses Three Way Calling, A cannot transfer the call because the message (i\_inform, B, A, “originator”) cannot be handled in the features states of TWC.
58. **TWC<sup>A</sup> & CT<sup>B</sup>**: if A is in a call resulting from a Call Transfer by B, A can only use TWC if A was the caller to B in the original (basic) call. (Otherwise A is in state CT21 or CT23, in which the flashhook event is not enabled.)
59. **VM<sup>A</sup> & CT<sup>A</sup>**: A cannot use Call Transfer to connect a caller with the Voice Mailbox, in this case for retrieving messages. (VM is only triggered by ‘dial’ event from state BC3, not from CT2.)
60. **VM<sup>A</sup> & CT<sup>B</sup>**: If B attempts to transfer a call to a line with Voice Mail, a deadlock may occur because the message (i\_inform, A, B, “call minder”) cannot be handled in the states CT5 and CT15.
61. **CT<sup>A</sup> & CT<sup>B</sup>**: It is not possible for both sides to transfer the call at the same time because the neither the i\_notify message nor the “originator” message can be handled in the features states of CT (on the subscriber’s side).
62. **CT<sup>A</sup> & CT<sup>B</sup>**: A call that has been transferred once cannot be transferred a second time because the parties of the resulting call end up in CT21 and CT22, or CT21 and BC7. Either way, one of them can not perform a flashhook event or process i\_notify messages. (*Cf.* interaction 58)
63. **GR<sup>A</sup> & CT<sup>B</sup>**: If a call is transferred to a Group Ringing subscriber there can be two i\_notify messages to the (eventual) originator of the resulting call, the second of which cannot be handled. (The first i\_notify message puts the (eventual) originator in state CT23 or GR5, depending on which feature was integrated last. The states GR4, GR5, CT21 and CT23 have identical transitions<sup>5</sup>, but neither of them allows for the second i\_notify message.)
64. **GR<sup>A</sup> & CT<sup>A</sup>/CT<sup>C</sup>/CT<sup>D</sup>**: (†) (Recall that C and D are the additional phones in the ‘group’.) A call established via Group Ringing cannot be transferred, since at least one of the parties is in state GR4 or GR5, respectively, and therefore cannot process an i\_notify or i\_inform message.
65. **CFB<sup>C</sup>/CFB<sup>D</sup> & GR<sup>B</sup>**: the Group Ringing subscriber’s phone expects only i\_free or i\_busy messages in response to alerting the other two phones in the group. If one of them also carries CFB and is busy when B receives an alert message, it will respond with an o\_notify message which B is not be able to process.

---

<sup>5</sup>NB: after the final bugfix for Group Ringing.

66. **CNDB<sup>A</sup> & GR<sup>B</sup>**: If a call from A to B is diverted (by Group Ringing) to C or D, A's identity is revealed to the terminating side. (*Cf.* interaction 1 between CNDB & CFB.)
67. **CW<sup>C</sup>/CW<sup>D</sup> & GR<sup>B</sup>**: (†) If one of the group phones (C or D) subscribes to Call Waiting, and is in state BC7 or BC11 when B receives an `i_alert`, B will receive `i_inform` in response to alerting that phone; however the Group Ringing feature states do not allow that message.
68. **CW<sup>C</sup>/CW<sup>D</sup> & GR<sup>B</sup>**: Call Waiting is disabled in states GR4 and GR5, so C or D cannot use call waiting if they are engaged in a call established via Group Ringing.
69. **CW<sup>A</sup> & GR<sup>B</sup>**: Call Waiting is disabled in states GR4 and GR5, i.e. if the first call from, say A, ends up connecting to the CW subscriber (C or D), that phone cannot process a message (`i_inform`, A, C *or* D, "cwhold").
70. **RBWF<sup>C</sup>/RBWF<sup>D</sup> & GR<sup>B</sup>**: If one of the group phones (C or D) subscribes to Ring Back When Free, and is busy when B alerts it, B will receive `i_inform` ("ringback") in response; however the Group Ringing feature states do not allow that message.
71. **RBWF<sup>C</sup>/RBWF<sup>D</sup> & GR<sup>B</sup>**: RBWF is not triggered when a call ends which has been set up using Group Ringing. (*Cf.* interaction 31 between RBWF and TWC.)
72. **RC/SB & GR**: If a call is connected to C or D using Group Ringing, the no special billing will apply even though C or D may subscribe to RC or SB. The billing record that is produced is the same as for a (CFB) forwarded call. (The requirements for Group Ringing are not clear on how GR is intended to be used or how it is expected to be charged.)
73. **TCS<sup>B</sup> & GR<sup>B</sup>**: are mutually exclusive.
74. **TCS<sup>C</sup>/TCS<sup>D</sup> & GR<sup>B</sup>**: If one of the group phones (C or D) screens calls from the subscriber's phone (B), Group Ringing may deadlock, since the "screened" message cannot be handled in states GR1 and GR2. (It is hard to see why a subscriber would choose this combination of features but it is nevertheless possible.)
75. **TWC & GR**: When a call has been established using Group Ringing, TWC cannot be used by either party, since they are in GR4 and GR5, respectively, and in these states the flashhook event is not enabled.
76. **TWC<sup>A</sup> & GR<sup>B</sup>**: (†) A call from A to B will result in a deadlock if it is connected to one of the group phones (C or D), because in TWC feature states the message (`i_notify`, B, A, N1 *or* N2) cannot be processed.
77. **VM<sup>B</sup> & GR<sup>B</sup>**: Group Ringin will override Voice Mail on the subscriber's phone, since it changes the `i_alert` transition from BC1 in such a way that BC9 is never reached.
78. **VM<sup>C</sup>/VM<sup>D</sup> & GR<sup>B</sup>**: If A alerts B, and B alerts C and D, the Voice Mail feature is be triggered (unless one of the phones goes offhook first, or A stops alerting B). This results in an `i_inform` message which cannot be processed in states GR1 – GR3.

79. **GR<sup>A</sup> & GR<sup>C</sup>**: (†) Group Ringing cannot safely be ‘chained’, i.e. if a phone, say C, in A’s ringing group also subscribes to Group Ringing, a deadlock may result since A cannot process `i_notify` messages from C. Even if that was resolved, there is no provision to forward the notification correctly.

Scenario: Incoming alert from B to A, triggers an alert to C, which in turn triggers an alert to, say, phone E in C’s ringing group. If E goes offhook, it send an `o_connect` message to C, which in turn sends an `o_connect` to A. However, A now ‘thinks’ that the call is to be established between B and C, while C ‘thinks’ that A and E are the parties of the resulting call. Therefore there will be messages (`i_notify`, A, C, B) (at C) and (`i_notify`, C, A, E) (at A) which cannot be handled. To complete the call setup these messages would have to be translated to (`o_notify`, C, E, B) and (`o_notify`, A, B, E) respectively.

### List of abbreviations

**CFB** Call Forward on Busy

**CNDB** Calling Number Delivery Blocking

**CW** Call Waiting

**RBWF** Ring Back When Free

**RC** Reverse Charging

**SB** Split Billing

**TCS** Terminating Call Screening

**TL** Teen-Line

**TWC** Three-Way Calling

**VM** Voice Mail