Hybrid Solutions for Feature Interaction Detection and Resolution

Muffy Calder

University of Glasgow

Mario Kolberg

Evan Magill

Stephan Reiff-Marganiec

University of Stirling

Dave Marples

Global Inventures Inc

UNIVERSITY OF STIRLING

Context of Research

- Interaction Handling Techniques
  - Offline:
    - not suitable in context of legacy systems, deregulated market
  - Online:
    - information available at runtime too limited for resolution
- HFIG Project
  - 1998-2001: funded by EPSRC, Mitel, Citel
  - joint between Glasgow and Strathclyde (later Stirling) Universities
  - investigate combination of offline & online techniques

Aims and Objectives

- Detect and resolve feature interactions
  - in the presence of legacy systems
    - (fragile code, no reliable documentation)
  - in a deregulated market
    - (third party features, short development periods)
- Approach shall
  - be embeddable in legacy and new architectures
  - not require changes to features or legacy code
  - not require design time information
  - automatically detect and resolve interactions at runtime

Outline

- Types of Interactions
- Detecting Interactions at Run-time
  - Dave Marples PhD thesis
- Message-Centric Approach
  - Stephan Reiff-Marganiec PhD thesis
- User-Centric Approach
  - Mario Kolberg's PhD thesis
- Results
- Conclusions

Types of Interactions

- STI: Shared Trigger Interactions
  - more than one feature reacts to a trigger

→ ***Message-Centric Approach***

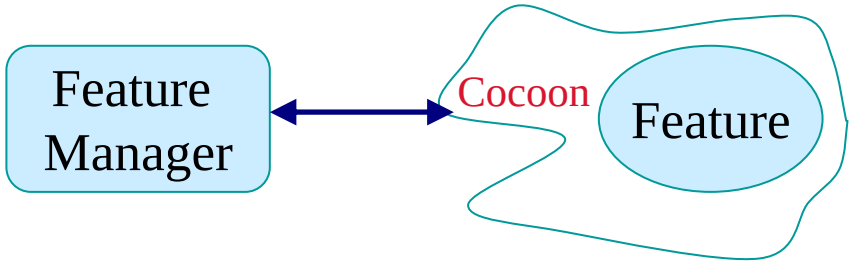
- SAI: Sequential Action Interactions
  - one feature's actions trigger another feature
- LI: Looping Interactions
  - special case of SAI's

→ ***User-Centric Approach***

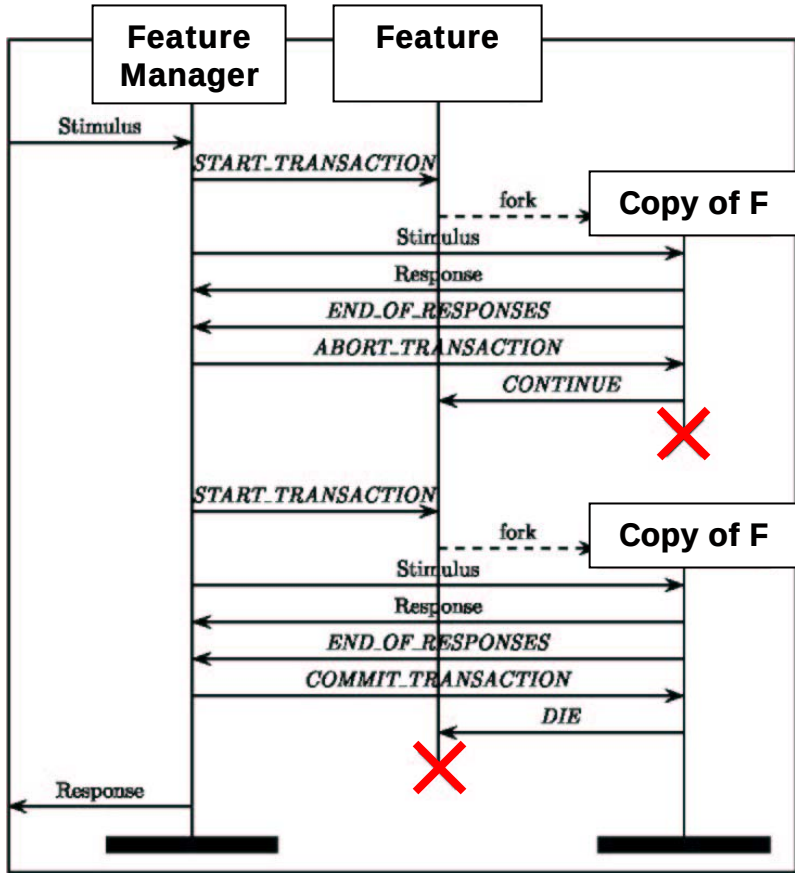
- MTI: Missed Trigger Interactions
  - one feature's actions prevent triggering another feature

Detecting Interactions at Runtime

- Features are embedded in a cocoon



- Transactional approach:
  - Commit and rollback
  - Copies of features



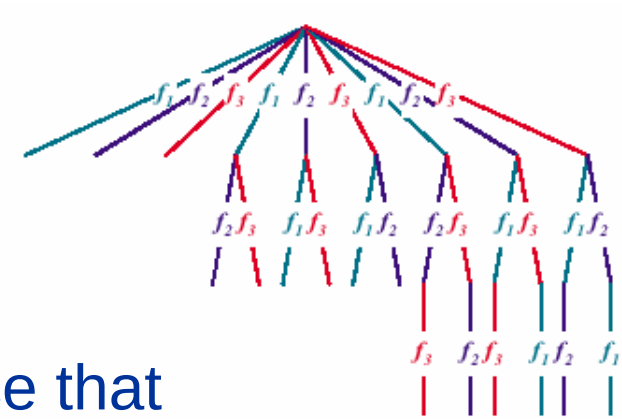
Message-Centric Approach

- Automatically selects good (if not best) resolutions
- Concentrates on handling STI's

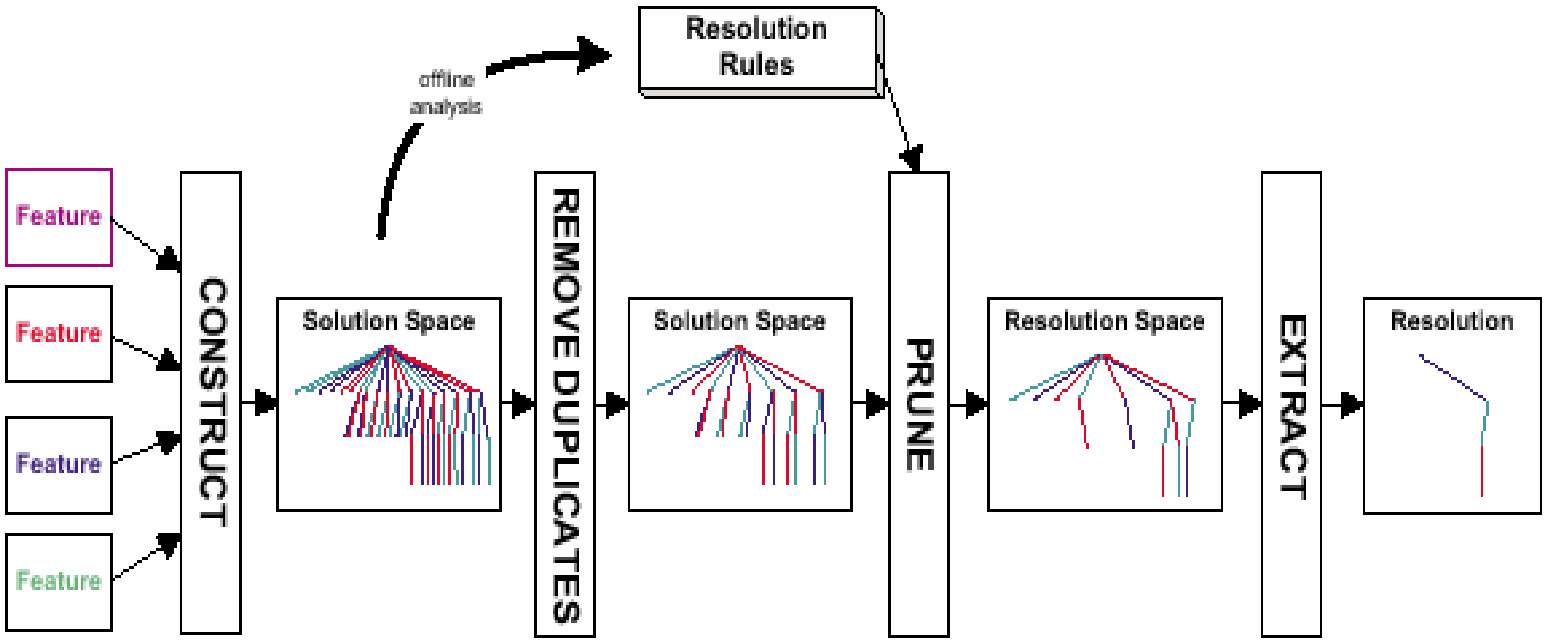
- FM constructs solution space as before
- Pruning and extraction allow to find resolutions
  - Guided by general rules
- Iterative improvement
  - Analyse solution space, define rules, analyse again, refine rules, ...

What are Solutions?

- Solution
  - a trace from one or more features running concurrently
- Solution space
  - the set of all solutions
- Resolution
  - a trace from the solution space that does not violate resolution rules
- Resolution space
  - the set of all resolutions



FM with Rule Based Resolution



Resolution

- Message Independent Rules
  - Duplicate subtrees sharing the same parent
  - Largest number of features
  - Highest priority
  - Choose one
- Message Dependent Rules
  - Classes of messages (announcements, tones, ...)
  - Regular expressions describing undesired behaviours

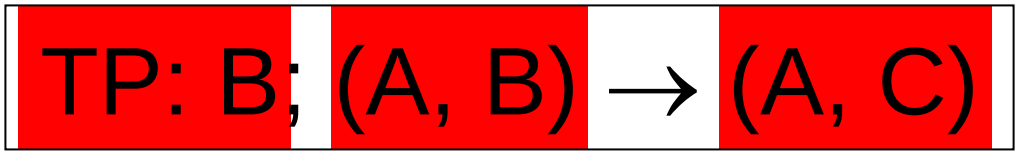
Example Resolution Rules

- Some rules in DESK
  - connecting a user to two different resources
  - routing to two different locations
  - routing a call away and changing user's state
  - routing a call away and connecting to resource
  - changing a user's state and connecting to a resource

User-Centric Approach

- Filtering approach
- Qualification of Sequential Action Interactions
- High-level view on connections
- Detects that certain features change behaviour as perceived by the user
- Simple algorithm
- Good run-time performance

Describing Features



- Triggering party
- Connection type
  - Source, destination
  - Original connection
  - Connection after feature activation
  - Parties & Treatment

Interaction Analysis

- Analysis pairs of features
- Compare two feature descriptions according to four rules

- Single User Dual Feature Control
- Connection Looping
- Redirection and Treatment
- Diversion and Reversing

An Example

POT 1

Initial Call Attempt, RtC is armed

POT 2

RtC, OCS(POT 1)

?

RtC initiates Callback, OCS blocks

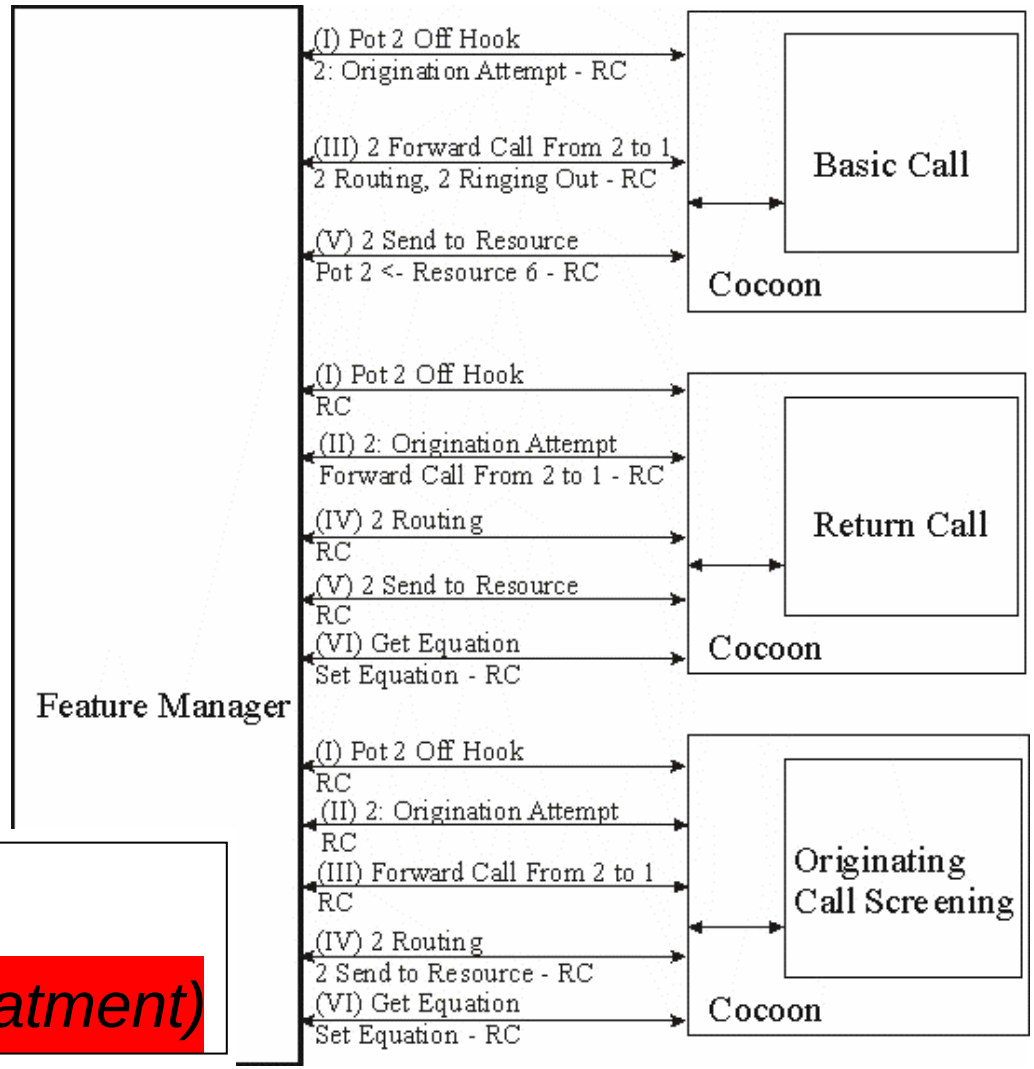
POT 1

POT 2

RtC, OCS(POT 1)

The Approach in Action

- Explore behaviour with on-line technique
- Cocoons
- If SAI detected → get connection equation
- Apply 4 rules



RtC: *TP: 2; (1, 2) → (2, 1)*

OCS: *TP: 2; (2, 1) → (2, Treatment)*

Single User Dual Feature Control

CFB: TP: B; (A, B) → (A, C)

CFU: TP: B; (A, B) → (A, C)

AR: TP: A; (B, A) → (A, B)

HL: TP: A; (A, B) → (A, B)

Connection Looping

CFB: TP: B; (A, B) → (A, C)

CFU: TP: C; (A, C) → (A, B)

Redirection and Treatment

CFB: TP: C; (A, C) → (A, B)

OCS: TP: A; (A, B) → (A, Treat)

AR: TP: B; (A, B) → (B, A)

OCS: TP: B; (B, A) → (B, Treat)

Diversion and Reversing

CFB:	TP: C;	(A, C)	→	(A, B)
AR:	TP: B;	(A, B)	→	(B, A)

CFB:	TP: A;	(B, A)	→	(B, C)
AR:	TP: B;	(A, B)	→	(B, A)

Results

	CFU	CW	CFB	OCS	TCS	VMS	RtC	ACB	DND	HL
CFU		M	M, U	U	M, U	M, U	M, U, U	M	M, U	
CW			M		M	M	M	M	M	
CFB				U	M, U	M, U	M, U, U	M	M, U	
OCS							U			U
TCS							M, U	M	M	U
VMS							M, U	M	M	U
RtC								M	M, U	M
ACB									M	
DND										U
HL										

- 10 features
- 49 interaction scenarios
- M → Message-Centric approach (STI, 28 cases)
  - Found “best” solution for all cases
- U → User-Centric approach (SAI, 21 cases)
  - Sometimes subjective decision

Conclusions

- Presented approaches
  - improve detection mechanism
    - qualification of interactions
  - add automated resolution
  - are complementary
    - each handles different class of interactions
- Future work
  - qualification of interactions into desired and undesired as perceived by user
  - application in other areas:
    - home networking, component based systems, IP telephony

Any Questions?