

Modular Feature Specification

Ken Turner
University of Stirling

4th April 2001



Introduction

- motivation:
 - separate description and analysis
 - formalisation in various languages
 - 'plug-in' domain and language
 - extensions for feature modularity
- notation:
 - root diagram
 - spliced feature
 - template feature
- toolset:
 - architecture
 - LOTOS:
 - framework
 - translation
 - analysis

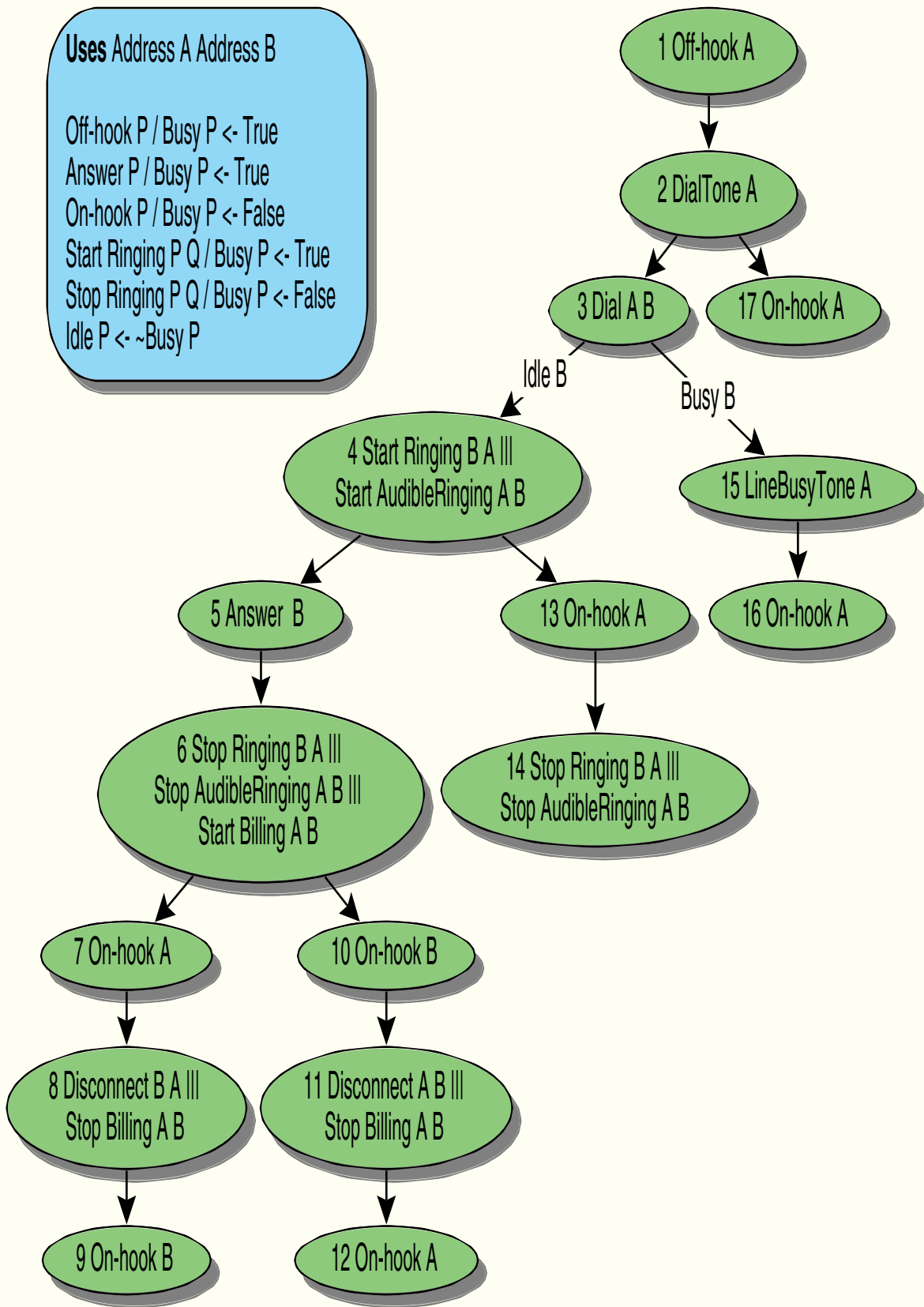
Motivation

- CRESS (CHISEL Representation Employing Systematic Specification) based on CHISEL:
 - mature industrial basis
 - simple graphical notation
- goals of CRESS:
 - separate description and analysis
 - backwards-compatible with CHISEL
 - defined, enforced diagram rules
 - fully formalised
 - modular, flexible feature description
 - 'plug-in' domain and language
- originally could handle:
 - features described in isolation
 - diagram translation to LOTOS and SDL
- extensions needed for:
 - self-contained features
 - flexible combination of features
 - multi-leg billing and redirection

Root Diagram – POTS

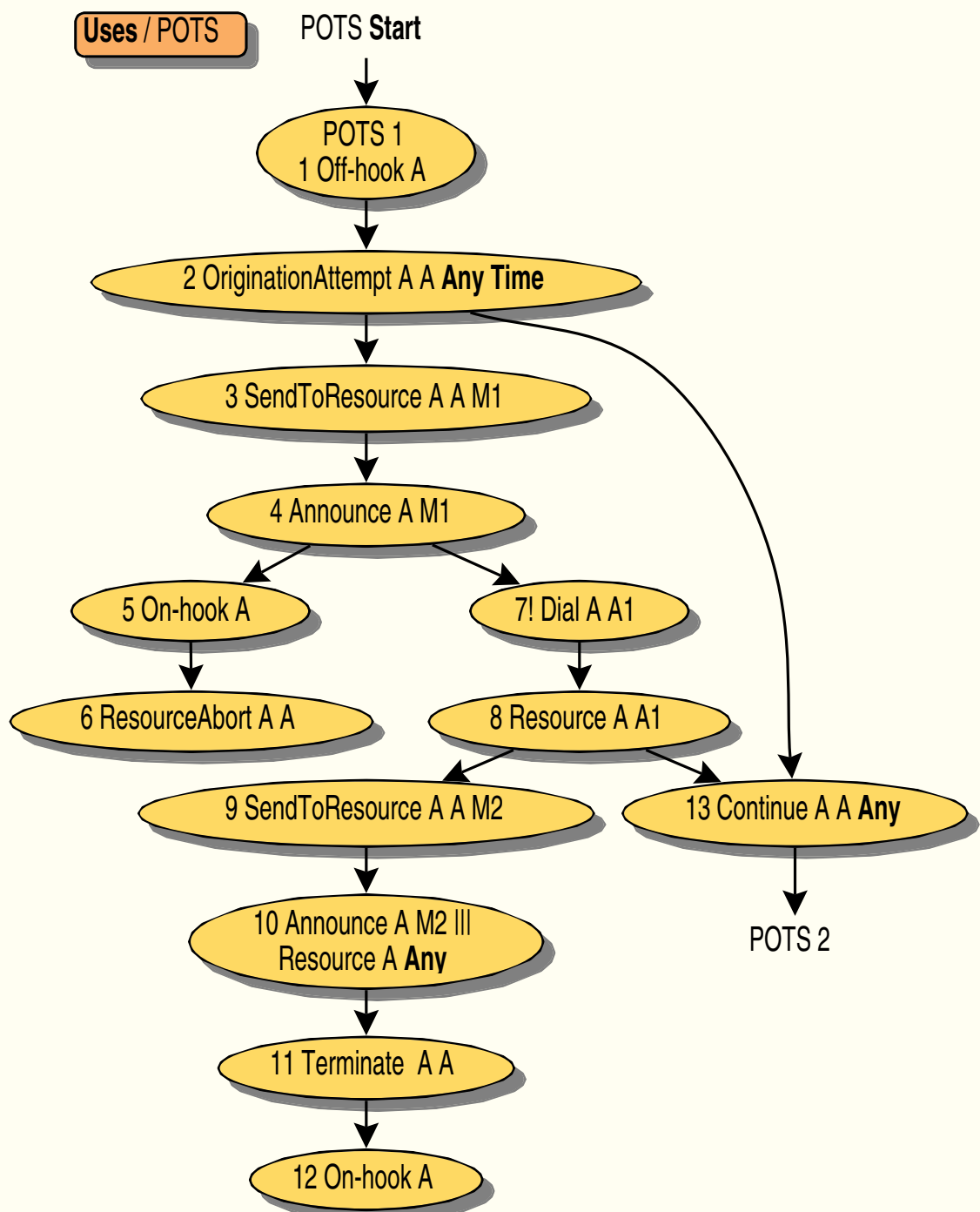
Uses Address A Address B

Off-hook P / Busy P <- True
 Answer P / Busy P <- True
 On-hook P / Busy P <- False
 Start Ringing P Q / Busy P <- True
 Stop Ringing P Q / Busy P <- False
 Idle P <- ~Busy P



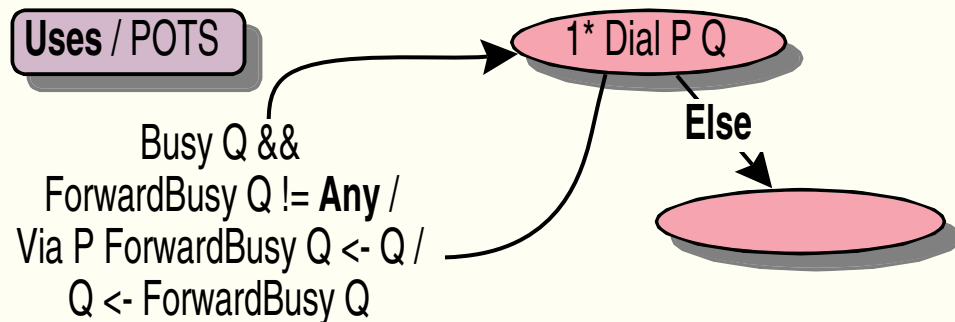
Spliced Feature

- splice a feature into root diagram if:
 - non-modular, non-parameterised
 - localised effect, applies only once
- INTL (IN Teen Line):

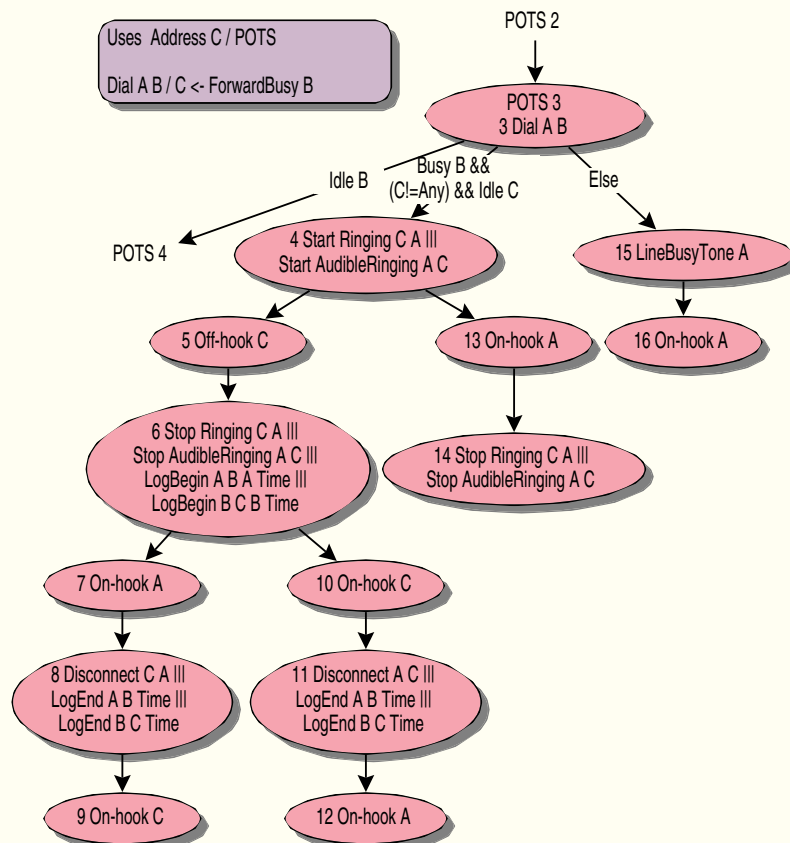


Template Feature

- instantiate feature in root diagram if:
 - modular, parameterised
 - spread effect, applies several times
- CFBL (Call Forward Busy Line):

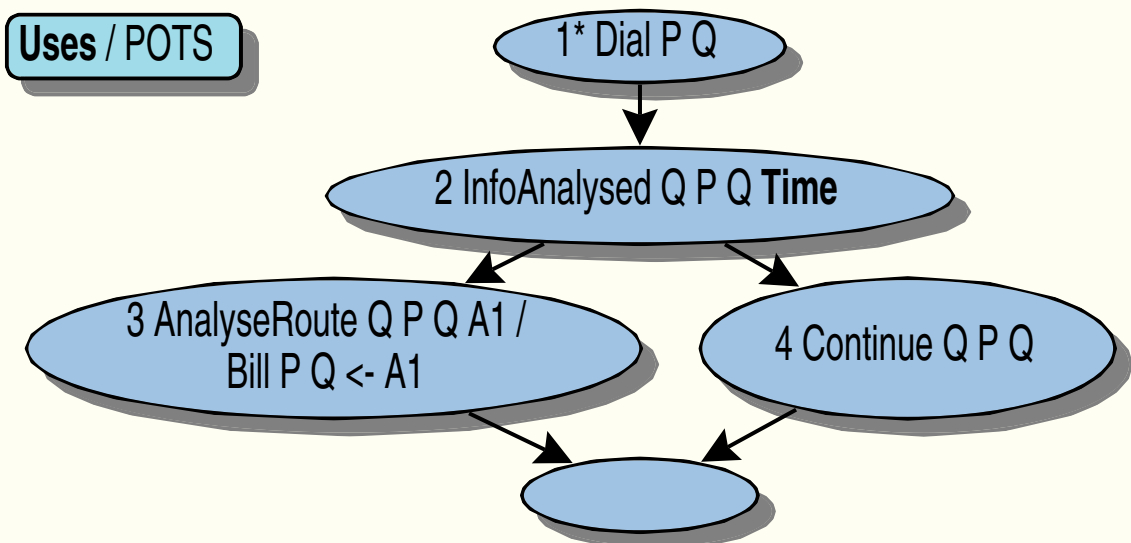


- compare this in spliced form:



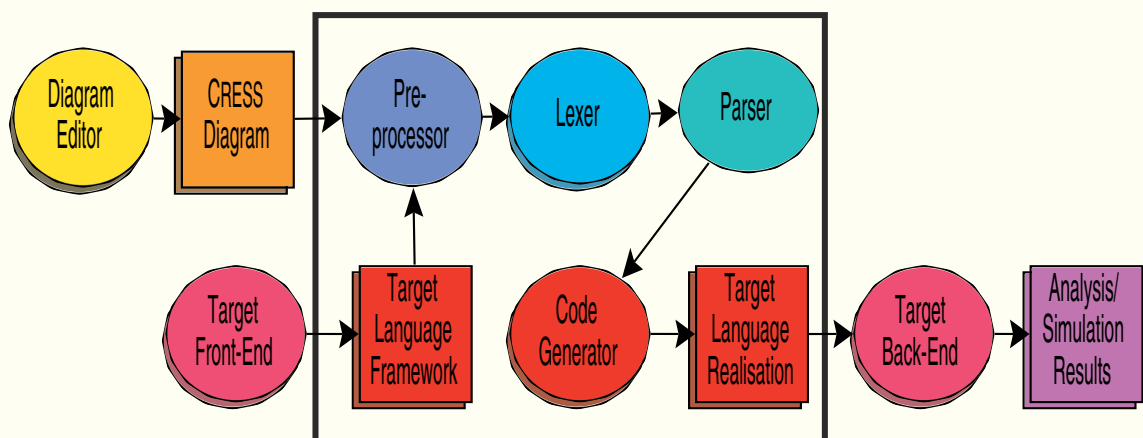
Billing and Redirection

- billing must be modelled properly:
 - raison d'être for operators!
 - can lead to interactions in its own right
 - billing variations needed (e.g. Credit Card, FreePhone, Split Charging)
 - forwarding leads to multiple billing legs
 - record each leg and who pays
 - chain features (e.g. INFB, INFR, INCF, CFBL)
 - template loops return to start of chain
 - *Start/Stop Billing* macro events
- INFB (IN Freephone Billing):



CRESS Tools

- designed to be independent of:
 - application domain
 - diagram editor
 - target language
 - platform
- framework depends on domain/language
- preprocessor instantiates:
 - domain-dependent data types
 - user profiles
 - root/feature combinations
 - for chosen target language
- toolset:



LOTOS Framework

Specification Network [User] : NoExit

Library ... (* library *)

Type ... (* pre-defined *)

Type ... (* domain-specific *)

Behaviour (* overall *)

Hide Bill,Stat,Scp **In**

CallInstances [Bill,Scp,Stat,User]

|[Scp]|

ServiceControl [Scp,Stat]

|[Stat]|

StatusManager [Bill,Stat] (...)

|[Bill]|

BillingSystem [Bill]

CRESS	LOTOS
diagram param	process param
Any	dummy / AnyAddress
node	direct / process call
event event	‘;’ / ‘ ’
event	event offer
event param	‘!’ / ‘?’
variable read	expression / read status
variable write	Let / write status
Time	read clock
leaf node	Stop / recursion

LOTOS Analysis

- emphasis so far on generic description
- validation strategy:
 - evaluate features in isolation
 - build up 'use case' scenarios
 - represent using ANTEST (ANISE Test)
 - check scenarios with all features
 - interactions show up as deadlock or non-determinism
- future plans:
 - techniques of others (e.g. Ottawa)
 - automated test generation
 - symbolic checking

Conclusion

- formalisation:
 - diagram rules thoroughly checked
 - automated translation to LOTOS/SDL
 - embedded in specification framework
- feature description:
 - spliced features for 'one-offs'
 - template features much more modular
 - multi-leg billing/redirection handled
- toolset:
 - domain/language/platform independent
 - early work on interaction analysis
- future work:
 - new telecomms uses (SIP, policies)
 - non-telecomms uses
 - automated test generation
 - symbolic checking

Discussion Points

- syntactic, not semantic, composition needed?
- representation more important than analysis?
- proper feature architecture can reduce need for interaction detection?
- diagrammatic, not symbolic, notation needed?
- validation more practical than verification?
- off-line analysis more useful than on-line resolution?
- what are modular features?
- why has 'BellCore' interest in CHISEL waned?
- how to get companies using CRESS?
- necessary to model speech in features?