Representing New Voice Services and Their Features

Ken Turner

University of Stirling

www.cs.stir.ac.uk/~kjt/research/cress.html

11th June 2003

CRESS
(Chisel Representation Employing Systematic Specification)

Introduction

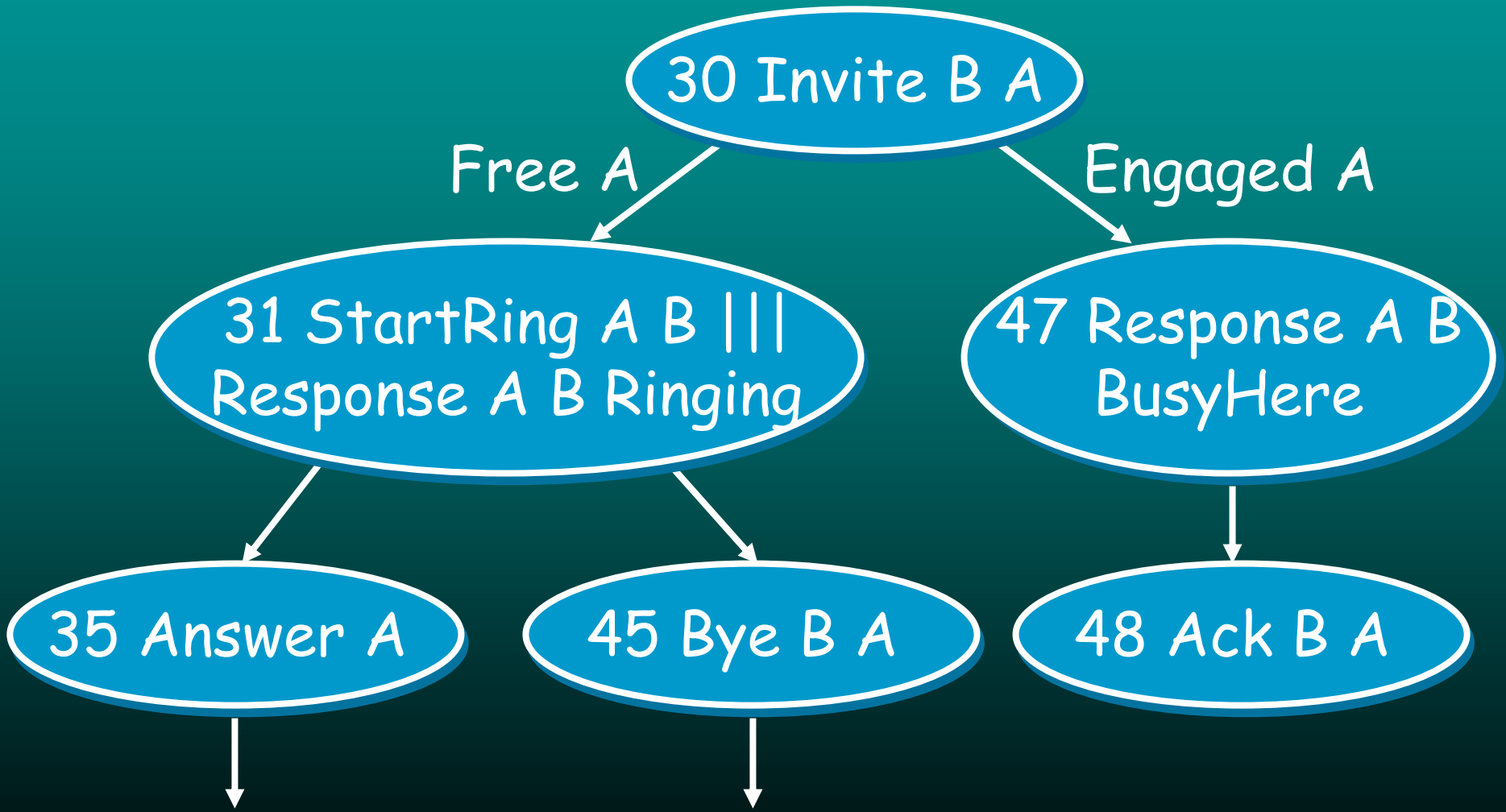
- flexible notation for feature description:
  - IN (Intelligent Network)
  - SIP (Session Initiation Protocol)
  - IVR (Interactive Voice Response)
- use for SIP (Internet Telephony):
  - service architecture
  - feature composition and interaction detection
- use for IVR (voice-activated systems):
  - structuring mechanisms
  - feature addition and interaction

CRESS Approach

- diagram types:
  - root (core behaviour)
  - feature (additional spliced/template behaviour)
- diagram elements:
  - nodes contain signals or actions (in parallel)
  - diagrams have sequences, branches, loops, calls
  - guards are expressions or events
  - rule boxes define transformations and macros
- automatic translation to:
  - formal languages (LOTOS, SDL)
  - implementation languages (Perl, VoiceXML)

CRESS Graph (SIP User Agent)

- graphical flow of input and output actions:



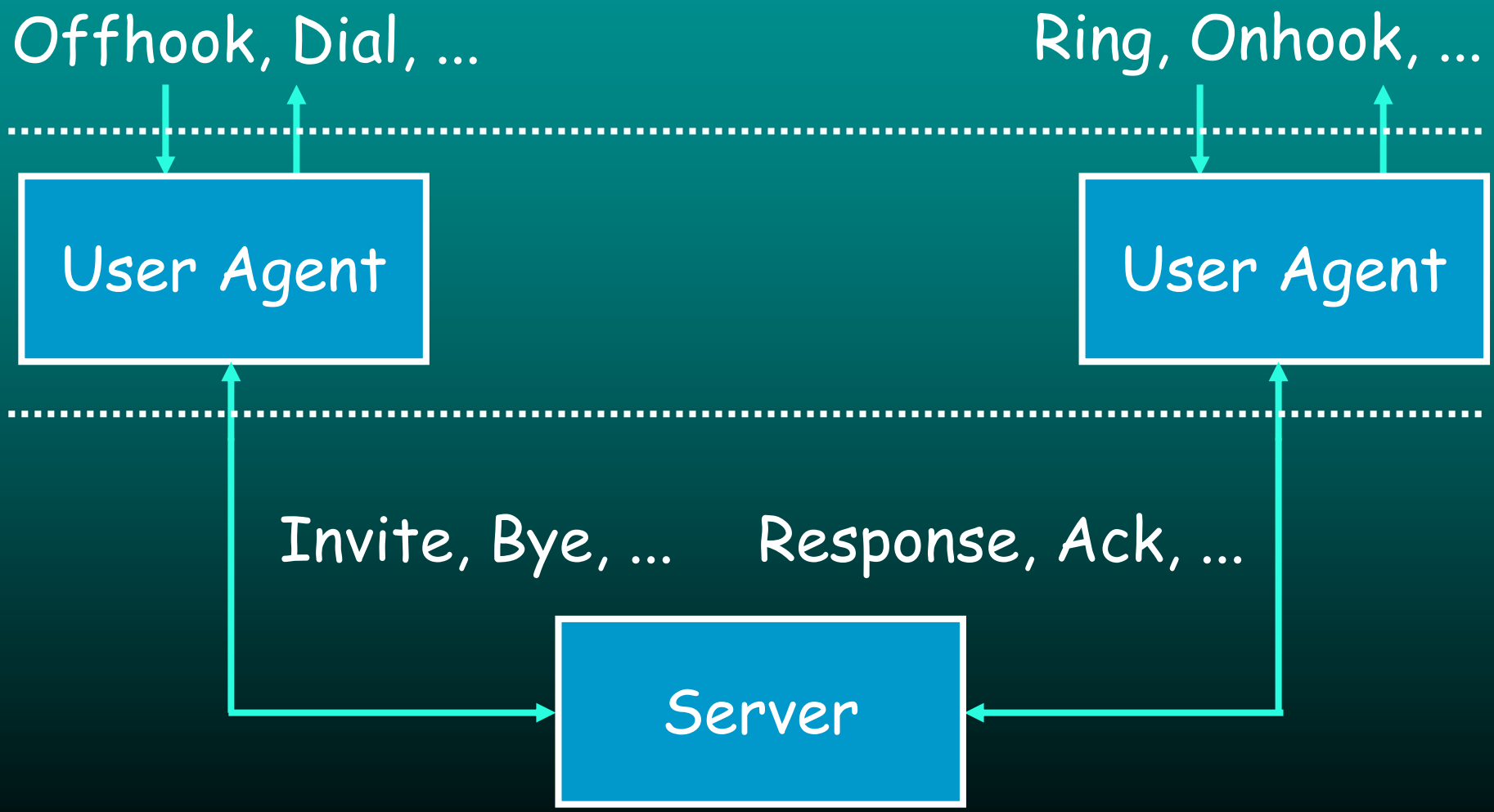
Feature Diagram

- adds behaviour:
  - spliced (cut and paste)
  - modular (instantiate and insert)
- feature may depend on:
  - root diagram
  - other feature diagrams
- complete behaviour is root plus features
- goals:
  - represent features graphically
  - support discovery of feature interactions

Application to SIP

SIP Model

- user↔protocol interface mapping:

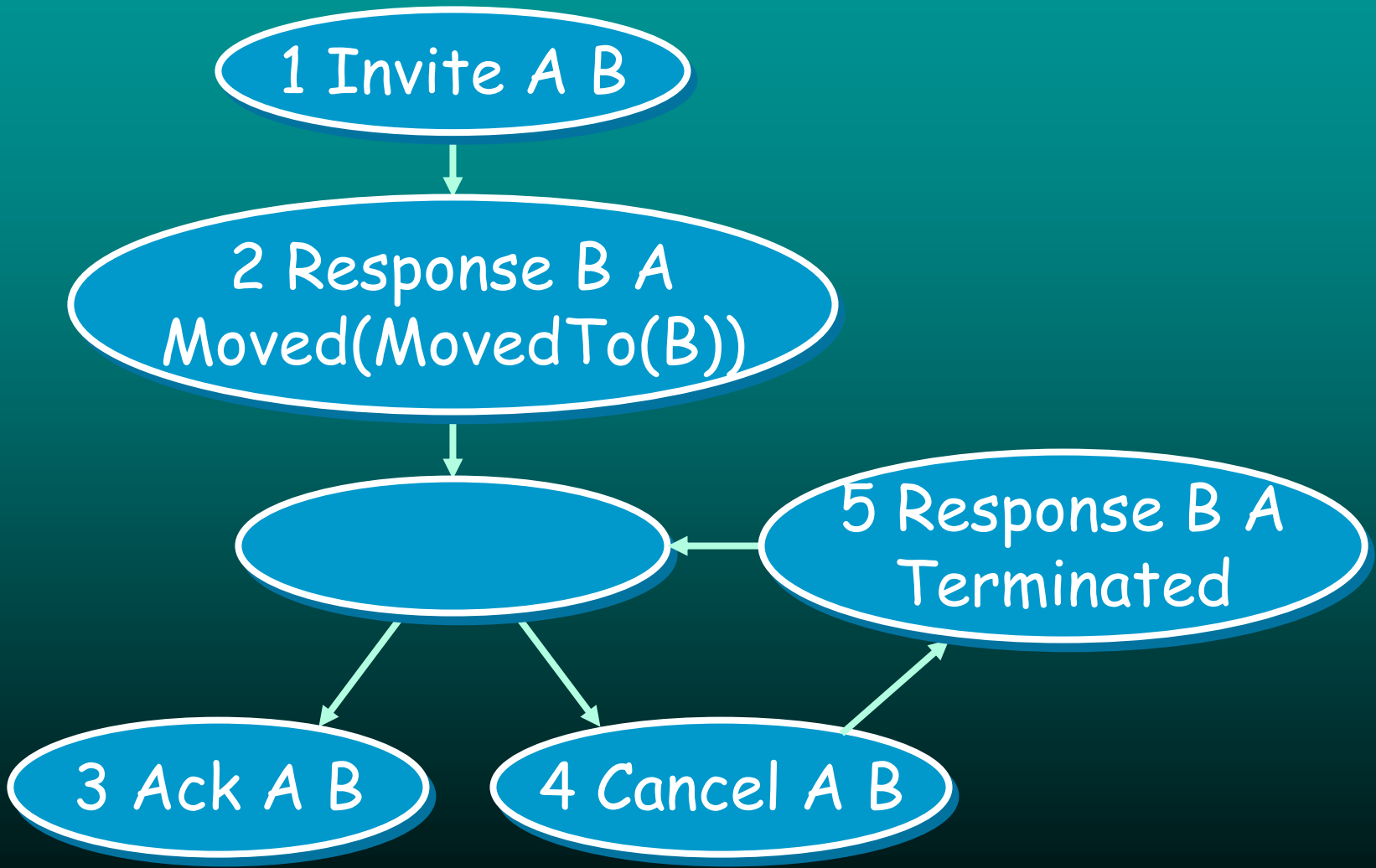


CRESS for SIP

- root diagram:
  - basic SIP call *or*
  - User Agent, Proxy Server, Redirect Server
- feature diagrams similar to IN
- automated analysis and simulation:
  - LOTOS
  - SDL
- translation to SIP scripts:
  - CGI (Common Gateway Interface) — Perl
  - CPL (Call Processing Language) — partial

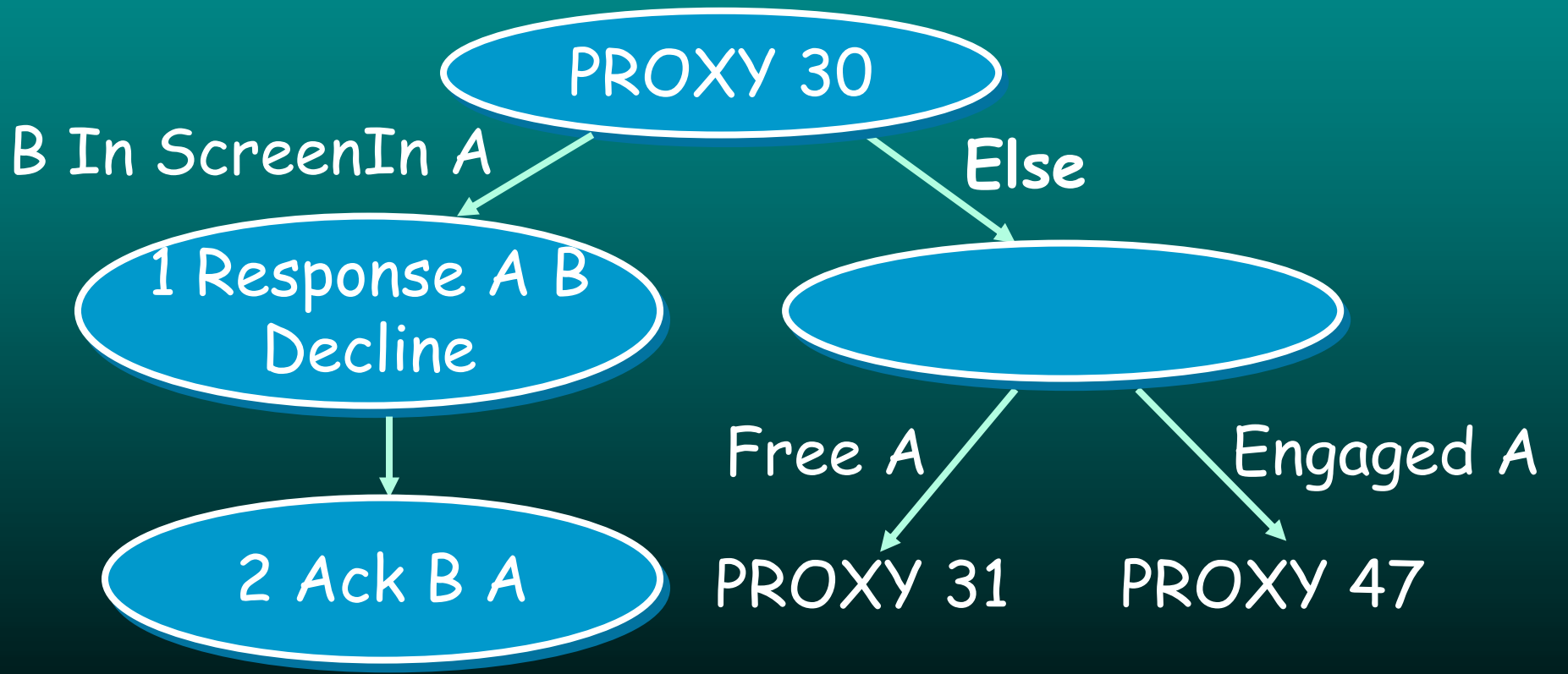
Root Diagram (Redirect Server)

- basic redirect service:



Spliced Feature (TCS)

- also defined as a modular feature
- pasted into Proxy Server diagram:



SIP Feature Study

- goals:
  - check integrity of feature descriptions
  - check features for mutual compatibility
  - smooth path to feature realisation (scripting)
- features validated individually:
  - generated specification simulated step-by-step
  - use-case scenarios
  - automated state exploration
- features validated in combination:
  - scenarios should still be valid
  - validation failure means feature interaction

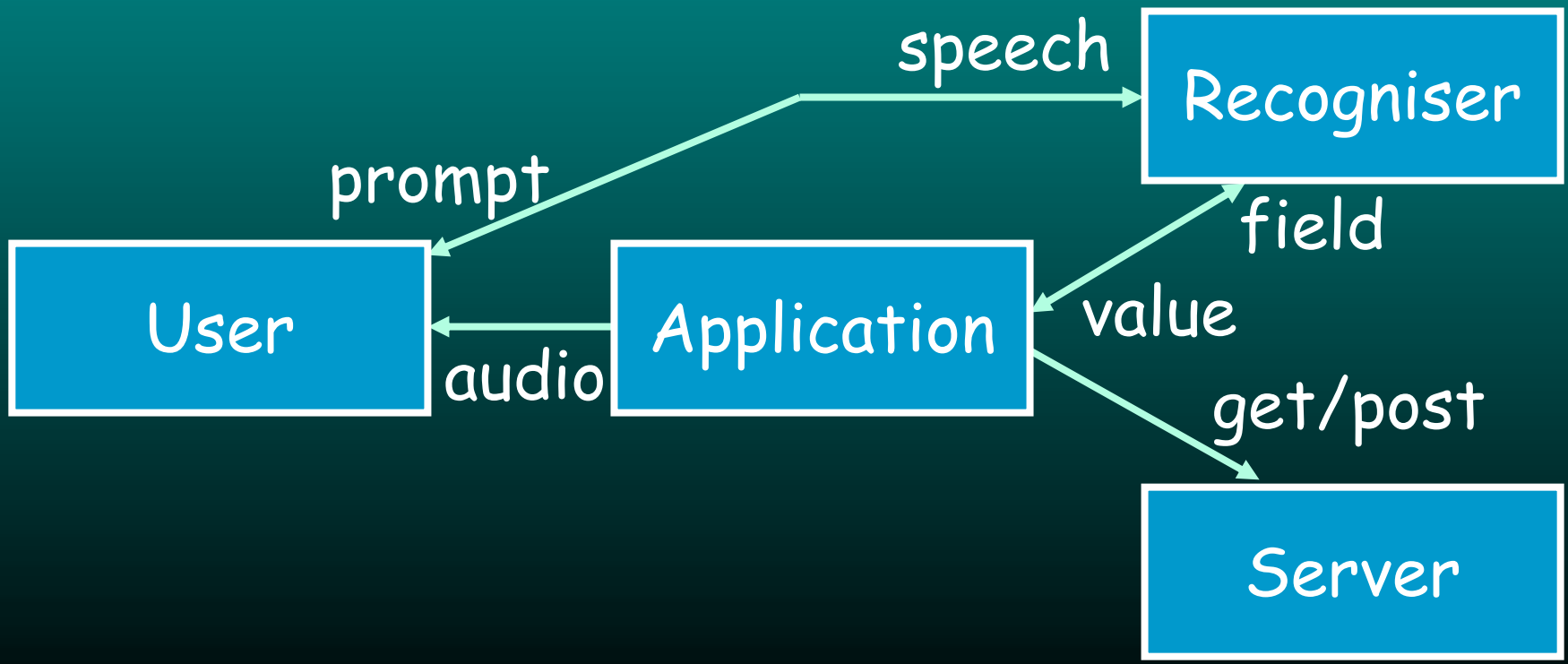
SIP Feature Results

- focus on feature representation:
  - busy is *defined*
  - IN-like features (forwarding, screening, …)
  - open to new kinds of SIP-specific features
- feature interaction detection separate:
  - interaction = behaviour change on combination
  - any LOTOS or SDL based technique can be used
  - conventional IN interactions can be shown
  - busy-based features may not interact
  - charge-based features may be irrelevant

Application to IVR

IVR Model

- form fields completed using speech input
- prompts from TTS (Text To Speech) or pre-recorded voice
- speech input directed by grammars



CRESS for IVR

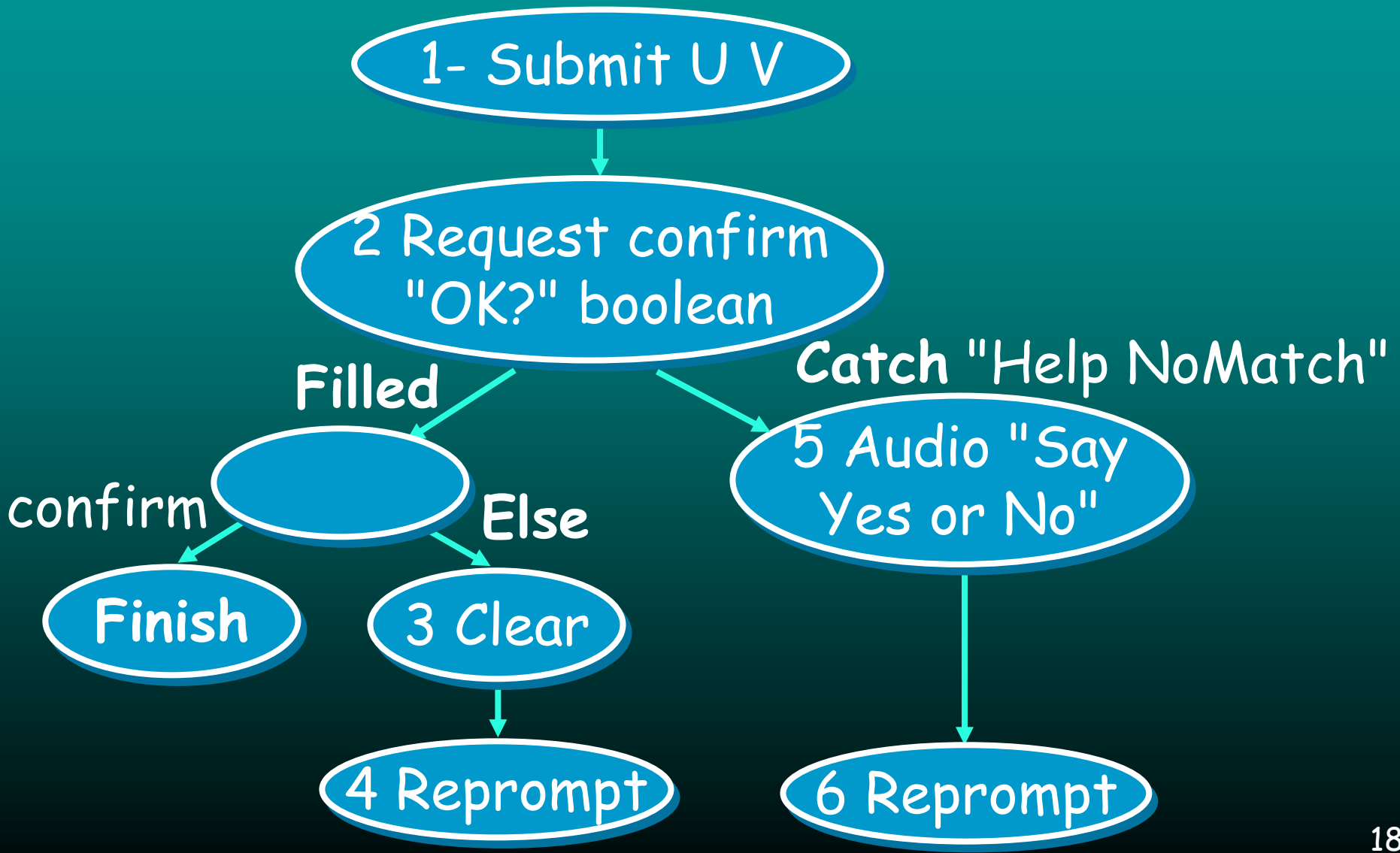
- root diagram:
  - defines core of IVR application
  - different for every application
- feature concept new:
  - features of general use
  - features for classes of applications
- automated analysis and simulation:
  - LOTOS
  - SDL
- implementation as IVR scripts:
  - VoiceXML (Voice Extended Markup Language)

Root Diagram (Product Order)

- basic ordering service:

Modular Feature (Submit Confirm)

- instantiate on each Submit:



IVR Feature Study

- goals:
  - check integrity of IVR applications
  - interpret meaning of features for IVR
  - check features for mutual compatibility
- features validated individually:
  - as for SIP
- features validated in combination:
  - as for SIP

IVR Feature Results

- focus on introducing features:
  - nothing similar in VoiceXML
  - useful role found for features
- feature interaction detection separate:
  - as for SIP
- different kinds of interactions occur, e.g.:
  - overriding properties, handlers, grammars
  - inconsistent use of data
- checking overall application behaviour a useful side-effect

Conclusion

- CRESS:
  - graphical but rigorous notation
  - flexible and adaptable
  - demonstrated in various application domains
- SIP:
  - service architecture defined
  - range of features described and analysed
- IVR:
  - new structuring mechanisms
  - addition of features
  - new kinds of feature interaction