FROM ARCHITECTURE TO REQUIREMENTS* :

A TELECOMMUNICATIONS*

SUCCESS STORY

Pamela Zave

AT&T Laboratories—Research

Florham Park, New Jersey, USA

pamela@research.att.com

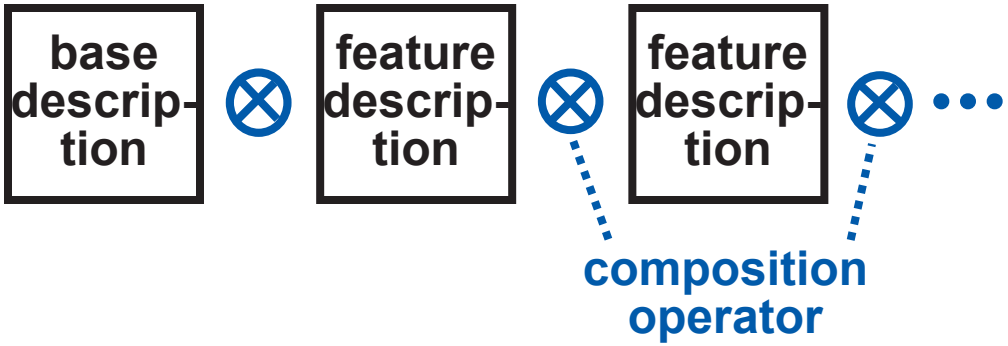
* This talk is about end-user requirements only.

* Telecommunications is networking with an emphasis on real-time communication among people.

FEATURES

A FEATURE is an increment, often optional, of functionality.

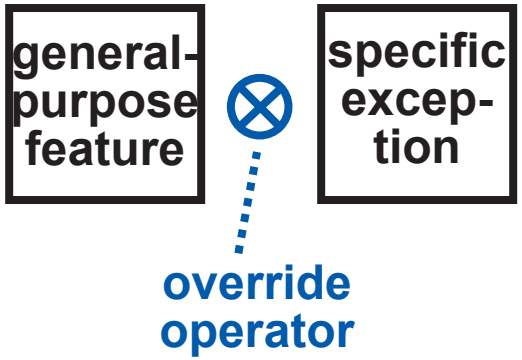
A FEATURE-ORIENTED DESCRIPTION:



FEATURE INTERACTIONS

A FEATURE INTERACTION is some way in which a feature modifies or influences another feature in defining overall system behavior.

for example:



feature interaction is an inevitable by-product of modularity in a feature-oriented description; it can be positive (desirable) or negative (undesirable)

THE FEATURE-INTERACTION PROBLEM

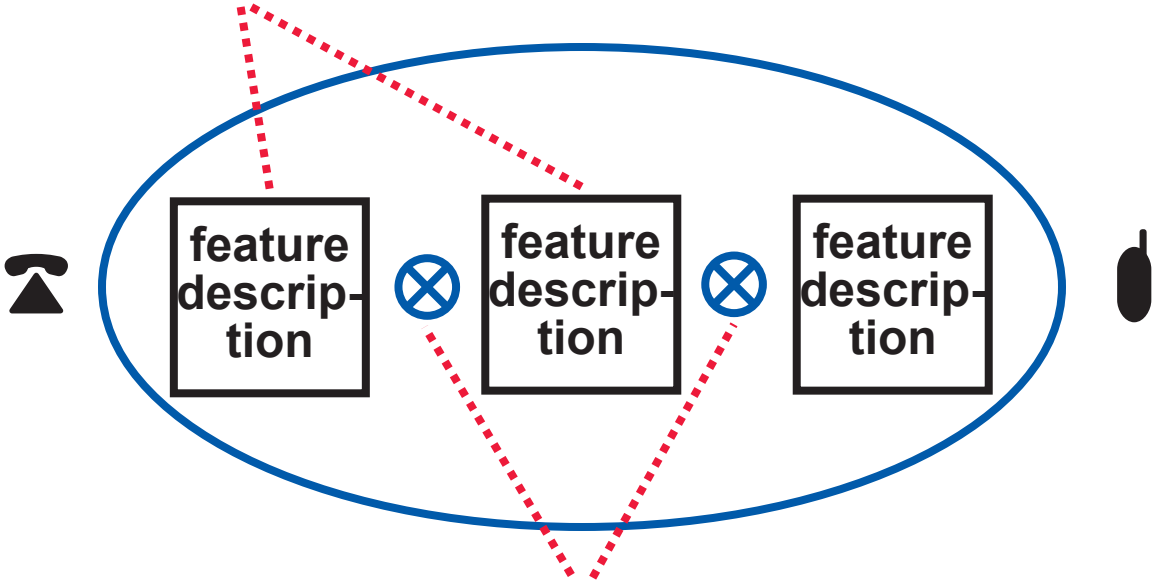
A feature-oriented description is easy to change, especially to change by adding new functionality, . . .

. . . but feature-oriented description makes feature interactions implicit, difficult to understand, and difficult to manage . . .

. . . which means preventing the bad ones and enabling the good ones.

TELECOMMUNICATION REQUIREMENTS OF TODAY

REQUIREMENTS FOR NEW, INDIVIDUAL FEATURES ARE TAKEN SERIOUSLY, CAN BE QUITE DETAILED



REQUIREMENTS FOR FEATURE INTERACTIONS ARE HAPHAZARD, LOCAL, USUALLY SUPERFICIAL

GLOBAL REQUIREMENTS (PROPERTIES, GUARANTEES) ARE MISSING ALTOGETHER
which is a major reason why requirements for feature interactions are poor

WHY NO GLOBAL REQUIREMENTS?

- the networks of today have been developing incrementally since the 1960s
- addresses, features, and other entities are highly ambiguous with respect to meaning and purpose
- users have conflicting goals
- there is little separation of concerns between requirements and implementation
- there are many interoperating networks

WHY ARCHITECTURE?

IN THE MID-1990s, NO PROGRESS ON TELECOMMUNICATION REQUIREMENTS SEEMED POSSIBLE

HOWEVER, INADEQUATE REQUIREMENTS WERE NOT THE ONLY SOFTWARE PROBLEM RELATED TO FEATURES:

productivity of the software-development organization for a large telephone switch:

1 line of code per meeting!

RECENTLY, MOST RESEARCH IN THIS AREA HAS BEEN ARCHITECTURE-ORIENTED

- agent architectures
- stack architectures
- Intelligent Network architectures

GOALS FOR TELECOMMUNICATION ARCHITECTURES:

modularity:

make it easy to add, delete, and change features

feature composition:

automatically eliminate many bad feature interactions, e.g., over-writing a variable

automatically enable many good feature interactions, e.g., forwarding invokes the features of the forwarded-to address

structured feature interaction:

constrain feature interactions

generality:

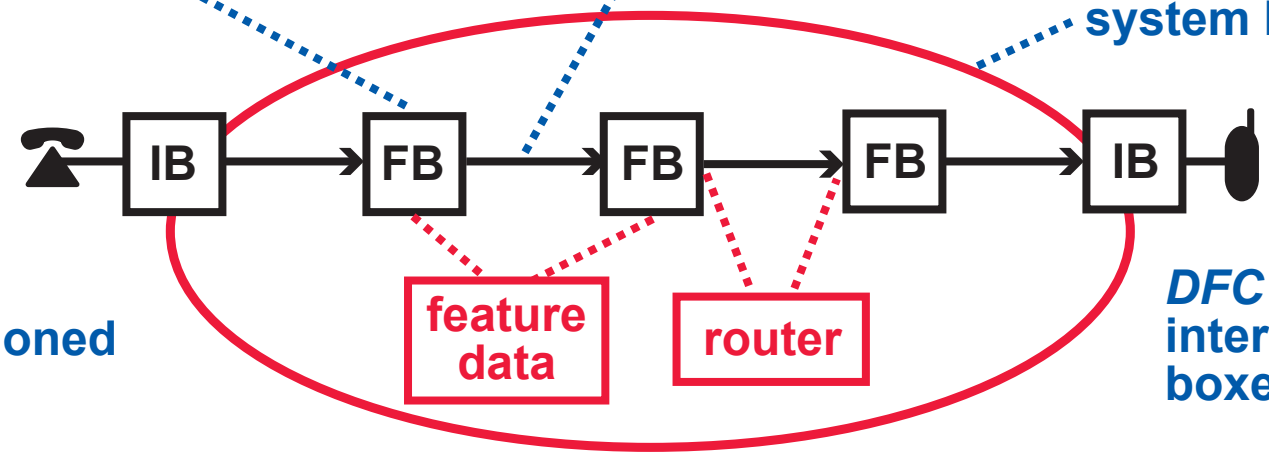
encompass all telecommunication services, present and future

DISTRIBUTED FEATURE COMPOSITION (DFC)

usage: a dynamically assembled graph of boxes and internal calls

box: a concurrent process, providing either interface or feature functions

internal call: a featureless, point-to-point connection with a two-way signaling channel and any number of media channels



feature data: usually partitioned by feature

DFC router: routes internal calls to boxes

FEATURE INTERACTION (COMPONENT COORDINATION) MECHANISMS:

two-way signaling along paths consisting of internal calls and intra-box *links*

the routing algorithm allows forks and joins, enables feature boxes to influence routing without knowing about others

THE MODULARITY MECHANISM IS PIPES AND FILTERS:

each box has transparency, autonomy, and context-independence

DFC WORKS!

HISTORY

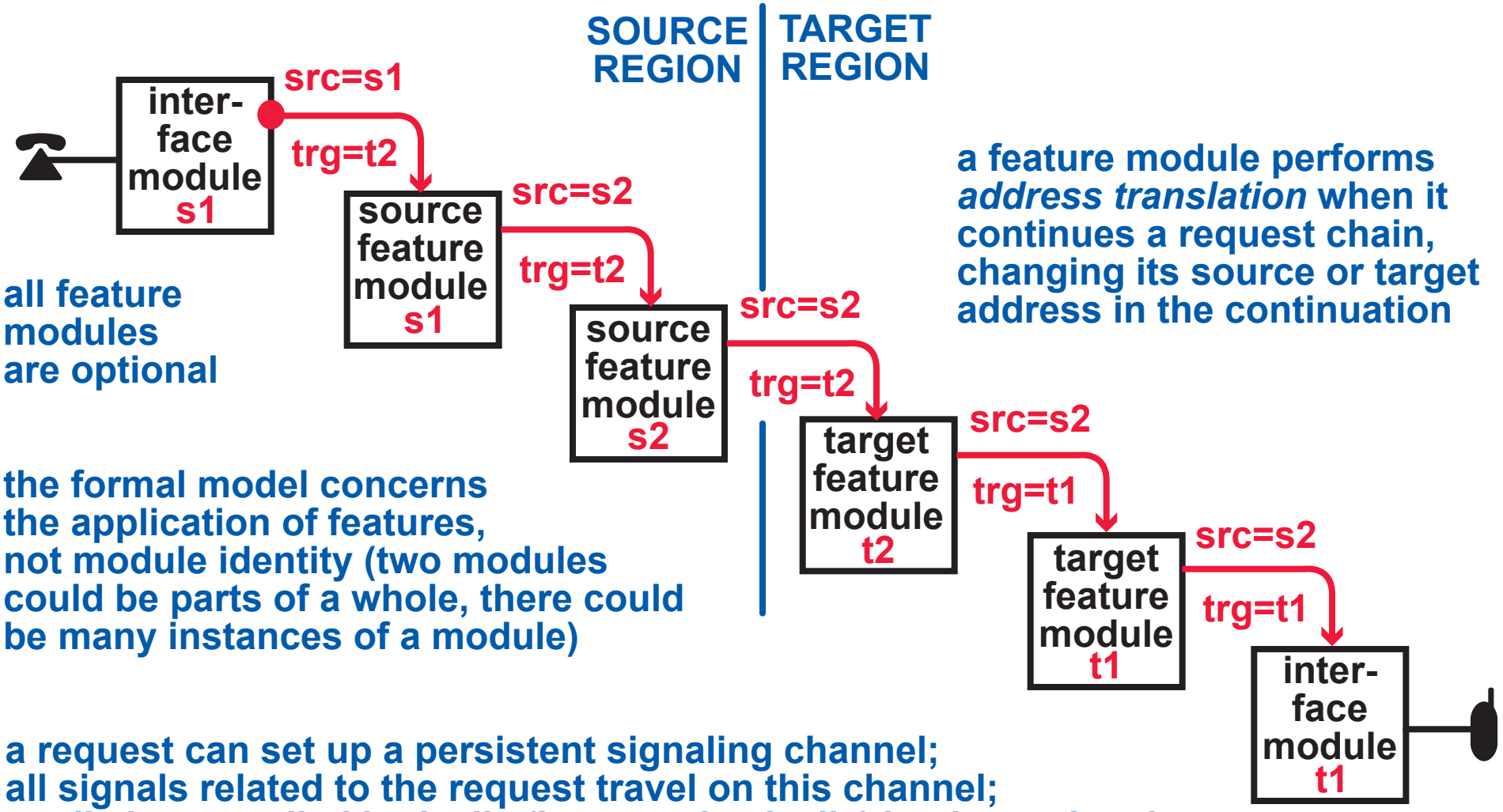
- the concept of DFC was originated by Michael Jackson and Pamela Zave 6 years ago
- work began on an IP implementation of DFC 4 years ago
- 1 year ago we began building voice-over-IP services for customers within AT&T
- we are a team of 8 people, plus additional contract programmers

ACCOMPLISHMENTS

- despite the penalty we pay for modularity, our performance is comparable to other voice-over-IP services, is improving steadily
- within AT&T, we have a reputation for making work what others can't make work
- at a recent trade show, we had the coolest demo
- in one year, we built an astounding variety of features (there was a lot of component and code re-use from earlier demos)
- we are at the forefront of standards work related to feature interaction in voice-over-IP
- we have had no trouble integrating Web services with our voice-over-IP services

FORMAL MODEL: REQUEST CHAINS

A TELECOMMUNICATION NETWORK CONNECTS DEVICES BY CREATING REQUEST CHAINS



all feature modules are optional

a feature module performs *address translation* when it continues a request chain, changing its source or target address in the continuation

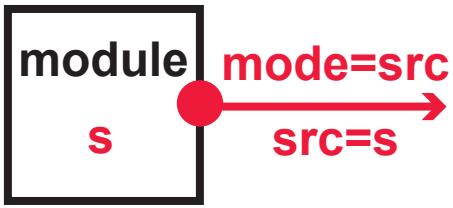
the formal model concerns the application of features, not module identity (two modules could be parts of a whole, there could be many instances of a module)

a request can set up a persistent signaling channel; all signals related to the request travel on this channel; media is controlled logically (but not physically) by these signals

any part of a signaling channel can be torn down at any time

FORMAL MODEL: ROUTING ALGORITHM

INITIATING MODULE



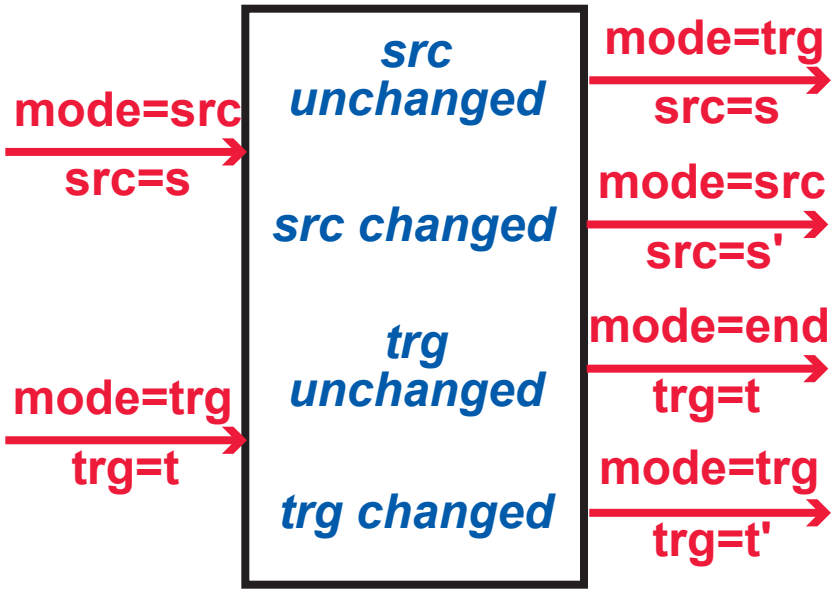
NETWORK ROUTER

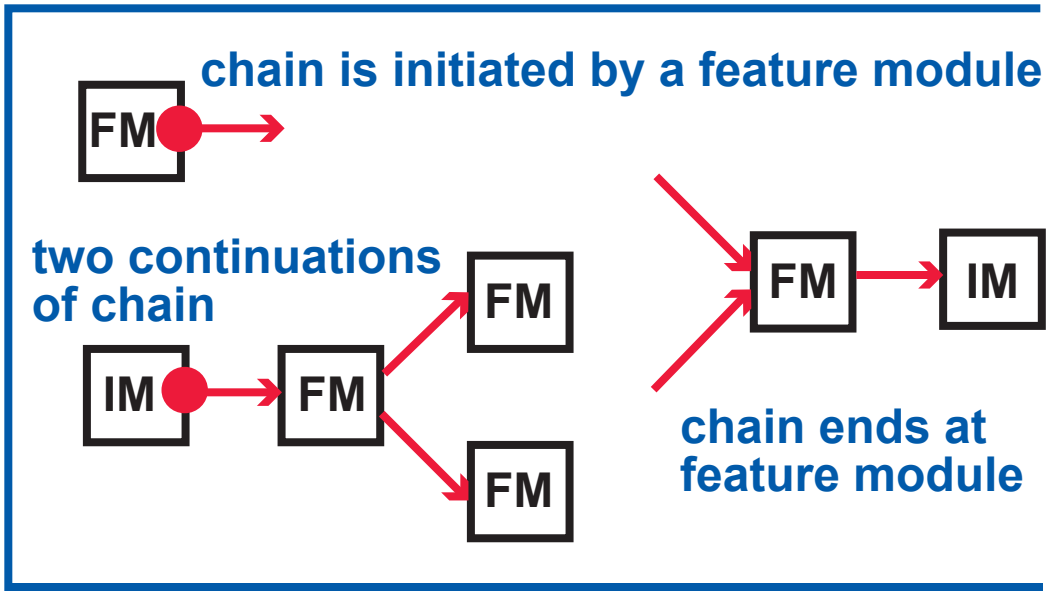
```
if (mode==src) then
   if (src has SFM m) then route to m
   else {mode:=trg; restart routing}

if (mode==trg) then
   if (trg has TFM m) then route to m
   else {mode:= end; restart routing}

else (mode==end)
   if (trg has IM m) then route to m
   else route to error module
```

CONTINUING MODULE





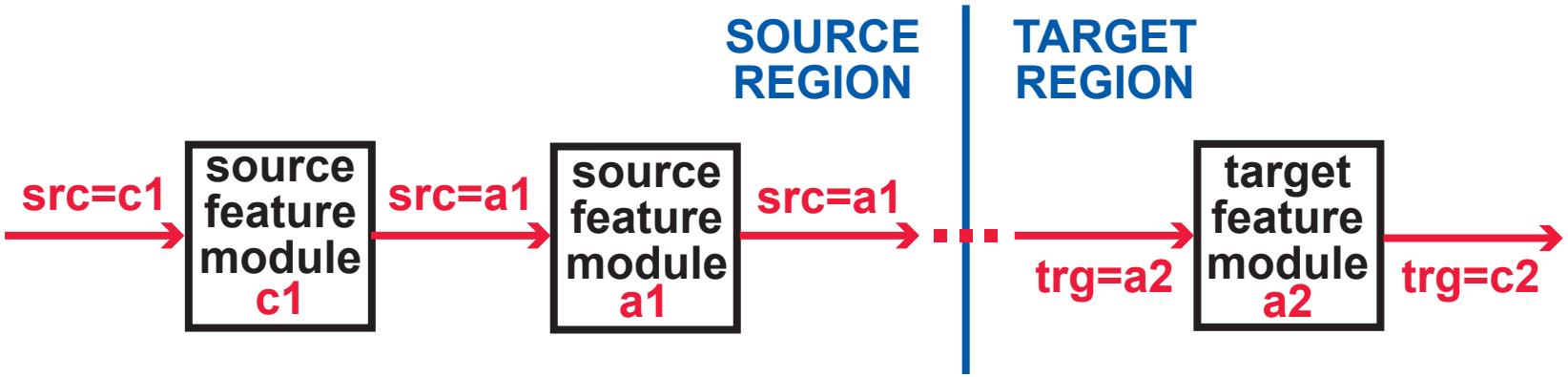
This is a simplifcation of DFC routing, to make the work more widely applicable.

ADDRESS-TRANSLATION FUNCTIONS

WHAT FUNCTIONS ARE BEING PERFORMED?

WHY ARE THEY BEING PERFORMED?

ON WHOSE BEHALF ARE THEY BEING PERFORMED?



if a1 and a2 identify:	then the source translation is:	and the target translation is:
groups	*affiliation:* affiliate the caller with the group	*representation:* find a representative of the group
mobile entities	*positioning:* position the mobile entity at the location of the calling device	*location:* find the location of the mobile entity
roles	*assumption:* assume the role for the caller	*resolution:* translate the role to the entity playing the role