FROM ARCHITECTURE TO REQUIREMENTS* :

A TELECOMMUNICATIONS*

SUCCESS STORY

Pamela Zave

AT&T Laboratories—Research

Florham Park, New Jersey, USA

pamela@research.att.com

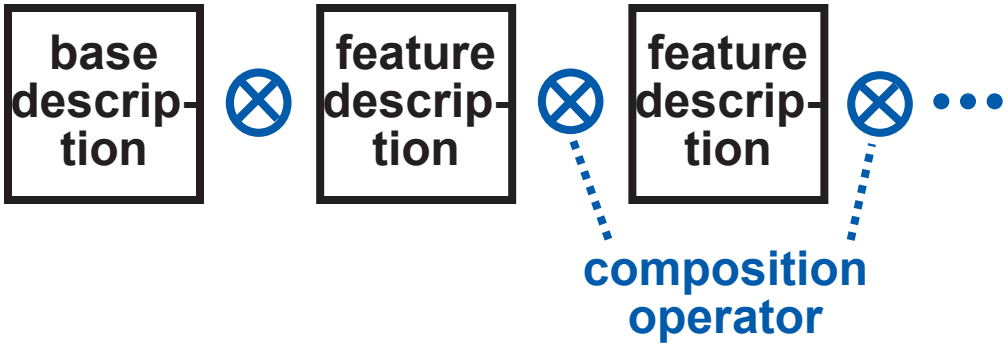
* This talk is about end-user requirements only.

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

FEATURES

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

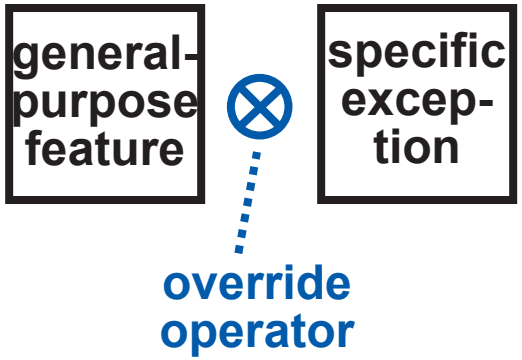
A FEATURE-ORIENTED DESCRIPTION:



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

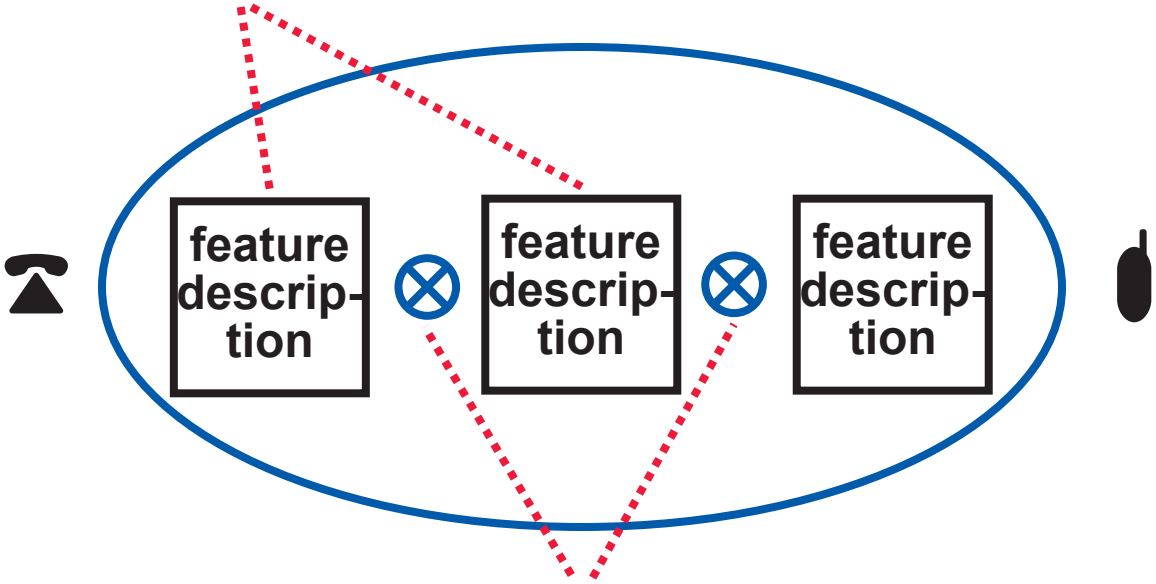
A feature-oriented description is easy to change, especially to change by adding new functionality, . . .

. . . 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:

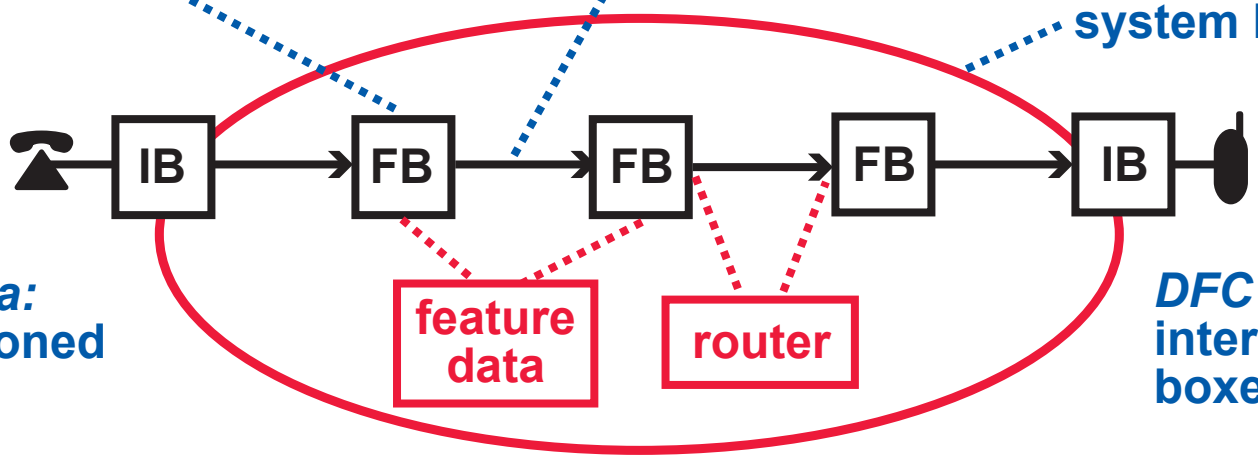
encompass all telecommunication services, present and future

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



persistent 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

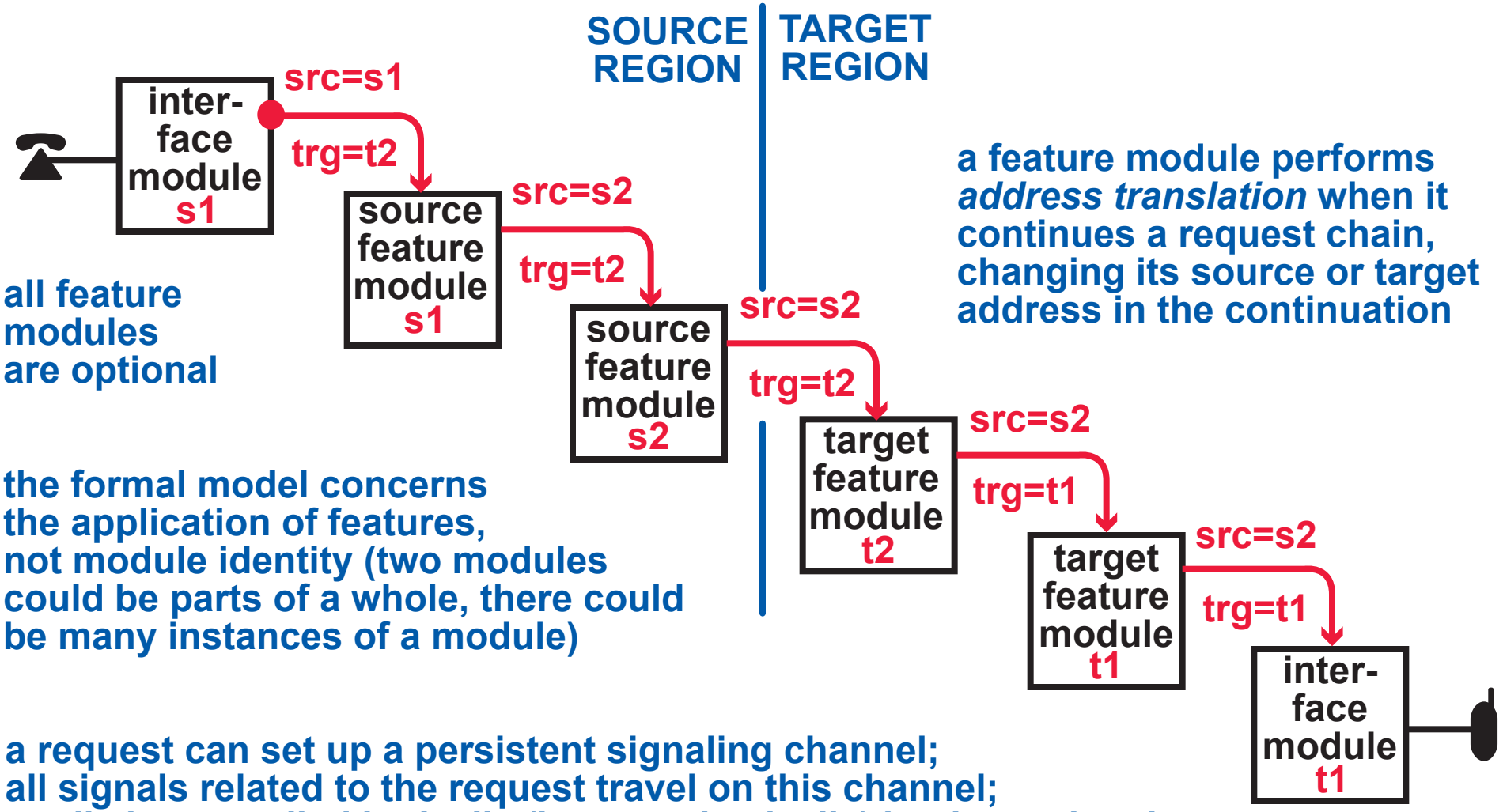
- the concept of DFC was originated by Michael Jackson and Pamela Zave 6 years ago
- work began on an IP implementation of DFC 4 years ago
- 1 year ago we began building voice-over-IP services for customers within AT&T
- we are a team of 8 people, plus additional contract programmers

ACCOMPLISHMENTS

- in one year, we built an astounding variety of features (there was a lot of component and code re-use from earlier demos)
- 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
- despite the penalty we pay for modularity, our performance is comparable to other voice-over-IP services, is improving steadily
- 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

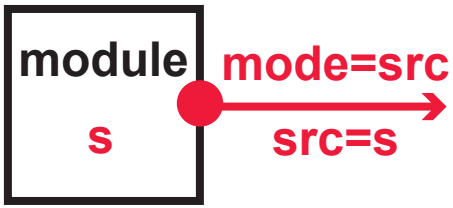
the formal model concerns the application of features, not module identity (two modules could be parts of a whole, there could be many instances of a module)

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



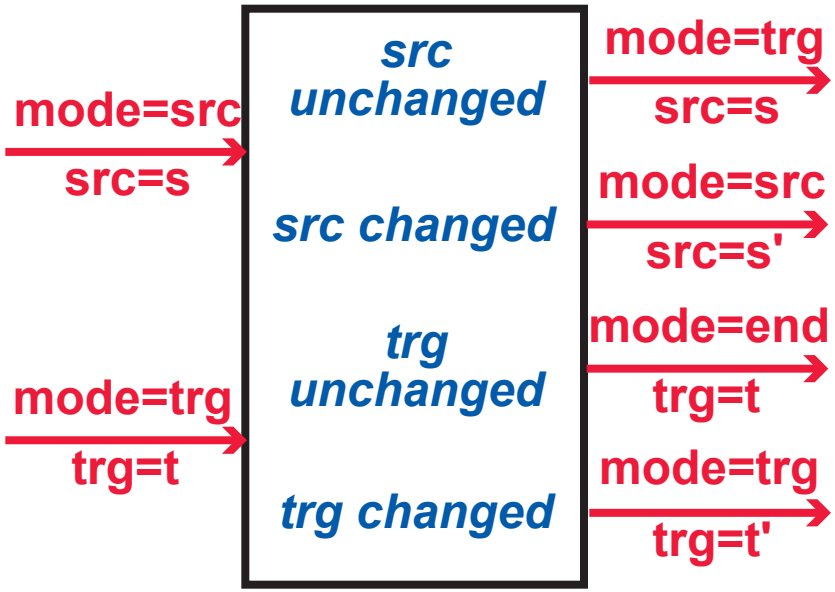
NETWORK ROUTER

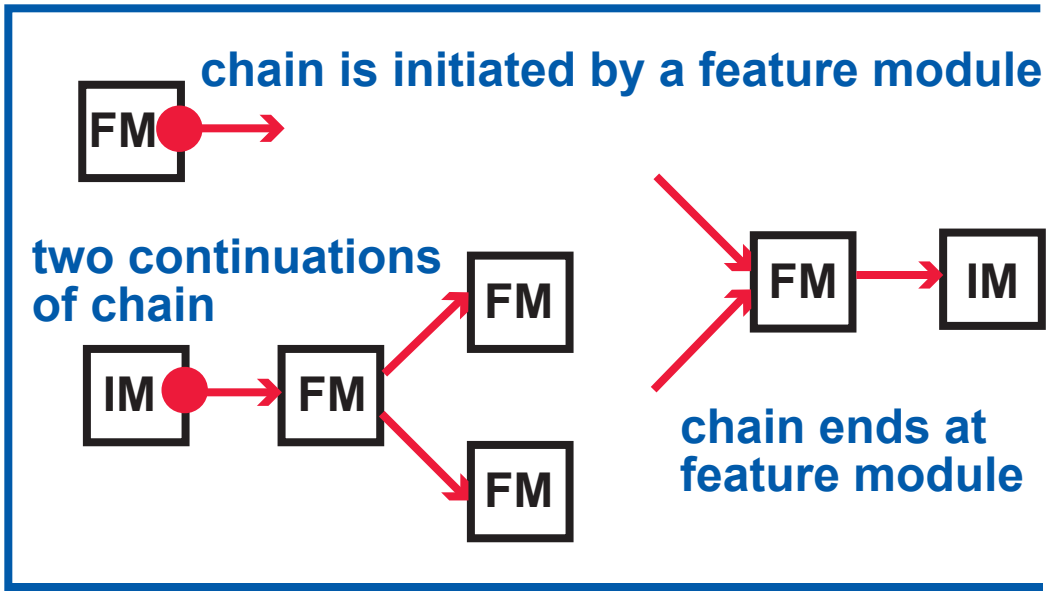
```
if (mode==src) then
   if (src has SFM m) then route to m
   else {mode:=trg; restart routing}

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





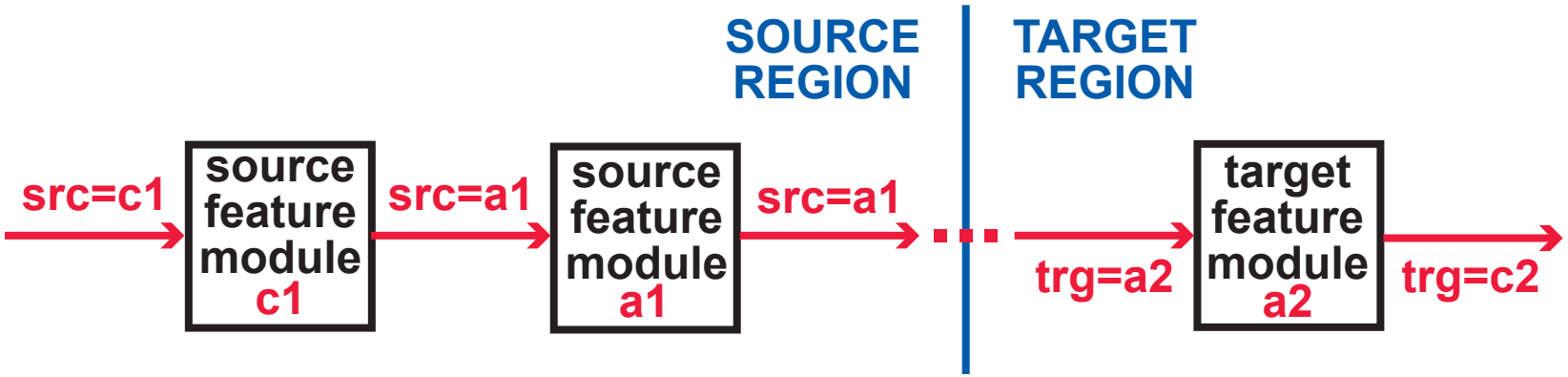
This is a simplifcation of DFC routing, to make the work more widely applicable.

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

ORGANIZATION OF ADDRESSES

EACH ADDRESS HAS ONE OR MORE OWNERS

- an owner has rights and responsibilities
- an owner knows the authentication secret

ADDRESSES MUST BE CATEGORIZED ACCORDING TO WHAT THEY IDENTIFY OR REPRESENT

for example:

- device
- person
- group
- role

and combinations thereof

ADDRESS CATEGORIES MUST BE PARTIALLY ORDERED BY "ABSTRACTION"

by definition:

- a group is more abstract than a person representing the group
- a person is more abstract than a device where he is located
- a public role is more abstract than a private identity

THE PRIMARY PURPOSE OF ADDRESS TRANSLATION IS TO CHANGE LEVEL OF ABSTRACTION

- in the source region, source addresses become successively more abstract
- in the target region, target addresses become successively more concrete

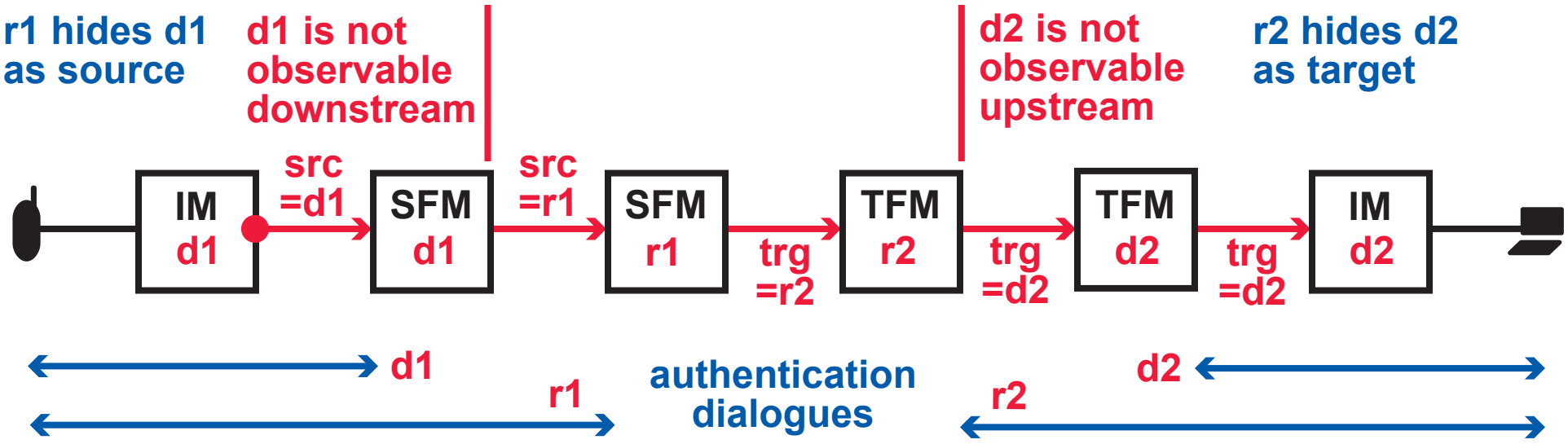
INTERACTION: IDENTIFICATION

PEOPLE AND FEATURE MODULES USE ADDRESSES TO IDENTIFY THE PARTIES WITH WHOM THEY ARE COMMUNICATING

A FEATURE THAT PERFORMS ADDRESS TRANSLATION INTERACTS WITH OTHER FEATURES BY AFFECTING THE IDENTIFICATION INFORMATION THEY RECEIVE

These principles balance conflicting goals:

PRIVACY	AUTHENTICITY
A person should be able to conceal a more private address that he owns behind a more public address that he owns.	A person should not be able to pose as an owner of an address he does not own.



INTERACTION: CONTACT

PEOPLE AND FEATURE MODULES USE ADDRESSES TO CONTACT THE PARTIES WITH WHOM THEY WISH TO COMMUNICATE

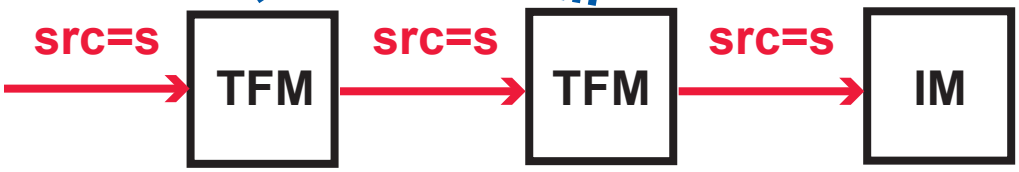
A FEATURE THAT PERFORMS ADDRESS TRANSLATION INTERACTS WITH OTHER FEATURES BY AFFECTING THE CONTACT INFORMATION THEY RECEIVE

REVERSIBILITY

A target feature module or callee should be able to call the source address of a request chain and and thereby target the entity that initiated it.

this is the most abstract source address, not the caller device

feature modules in the target region must not change the source address



REPRODUCIBILITY

A feature module or person should be able to call the same entity twice and be connected to the same representative of that entity.

conflicts with mobility and the freedom of representation functions

INTERACTION: INVOCATION

THE ADDRESSES IN A REQUEST CHAIN DETERMINE WHICH FEATURE MODULES ARE IN THE CHAIN

A FEATURE THAT PERFORMS ADDRESS TRANSLATION INTERACTS WITH OTHER FEATURES BY AFFECTING WHICH FEATURES ARE INVOKED

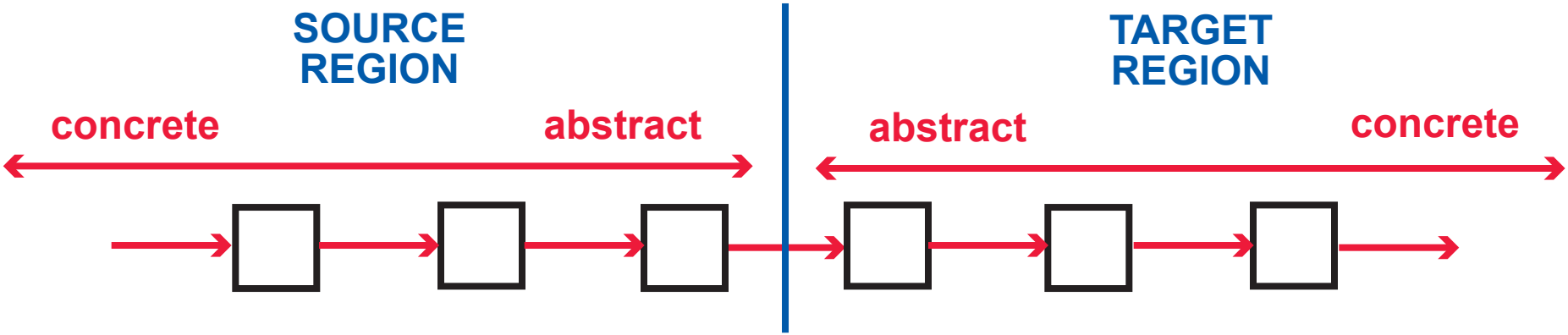
BOUNDEDNESS

The numbers of source and target feature modules in a chain should be bounded.

leads to

MONOTONICITY

In a region, the feature modules of more concrete addresses should be closer to the outer end of the region than feature modules of more abstract addresses.



each feature module knows where the more abstract and more concrete features are

features can be prioritized and coordinated (e.g., by token passing) without knowledge of other features

IDEAL ADDRESS TRANSLATION . . .

. . . IS A SET OF CONSTRAINTS . . .

Constraint 1: A target feature module in a request chain does not change the source address of the chain.

Constraint 2s: If a source feature module in a request chain translates the source address, the new source address is more abstract than the old one.

Constraint 2t: If a target feature module in a request chain translates the target address, the new target address is more concrete than the old one.

. . . THAT GUARANTEE PROPERTIES . . .

Source Privacy: If s1 is a source address in a request chain, and if s1 has a source feature module that changes the source address to s2 in this chain, then s1 is not observable as a source downstream of this module.

Target Privacy: If t2 is a target address in a request chain, and if t2 has a target feature module that changes the target address to t1 in this chain, then t1 is not observable as a target upstream of this module.

. . . BASED ON THE PRINCIPLES . . .

privacy *reversibility* *boundedness*
authenticity *monotonicity*

. . . IN A WAY THAT IS . . .

modular: modules do not cooperate explicitly with other modules, or know which modules are present

extensible: adding (or deleting) features does not require changing existing (or remaining) features

THE CONSTRAINTS OF IDEAL ADDRESS TRANSLATION ARE GLOBAL COORDINATING CONVENTIONS FOR TELECOMMUNICATION FEATURES

RELATION OF IDEAL ADDRESS TRANSLATION TO REQUIREMENTS ENGINEERING

THE PRINCIPLES OF PRIVACY, AUTHENTICITY, REVERSIBILITY, AND BOUNDEDNESS ARE "PROTO-REQUIREMENTS"

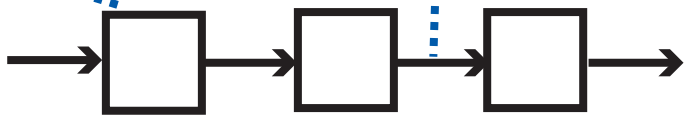
Privacy: A person should be able to conceal a more private address that he owns behind a more public address that he owns.

vague, informal

THE ARCHITECTURE IS FORMALLY DEFINED, STRESSES MODULARITY AND EXTENSIBILITY

modules can be added easily

each module is context-independent



there are no true requirements, satisfiable by systems with any architecture

this is not the only way that the goals could be achieved

without the clarity provided by the architecture, the principles would not have been discovered yet

formalized in terms of request chains

we know what concealment is (observable by module = observable by owner of module's address)

THE PROPERTIES ARE PRECISE AND FORMAL; THEY SATISFY THE PRINCIPLES IN A WAY THAT IS EASY TO UNDERSTAND, MODULAR, AND EXTENSIBLE

Source Privacy: If s1 is a source address in a request chain, and if s1 has a source feature module that changes the source address to s2 in this chain, then s1 is not observable as a source downstream of this module.

RELATION OF IDEAL ADDRESS TRANSLATION TO THE REAL WORLD OF NETWORKING

THERE ARE MANY REASONS WHY THE REAL WORLD MIGHT NOT CONFORM TO THE IDEAL

- inadequate infrastructure
- legacy of noncompliant features or address mappings
- interoperation with untrusted networks
- unwise optimizations
- one legitimate case in which a constraint is (deliberately) too strong

THERE ARE MANY WAYS TO COPE WITH THESE EXCEPTIONS

- refine or adapt the reasoning
- trace which properties do and do not hold
- enforce the constraints in a subnetwork only

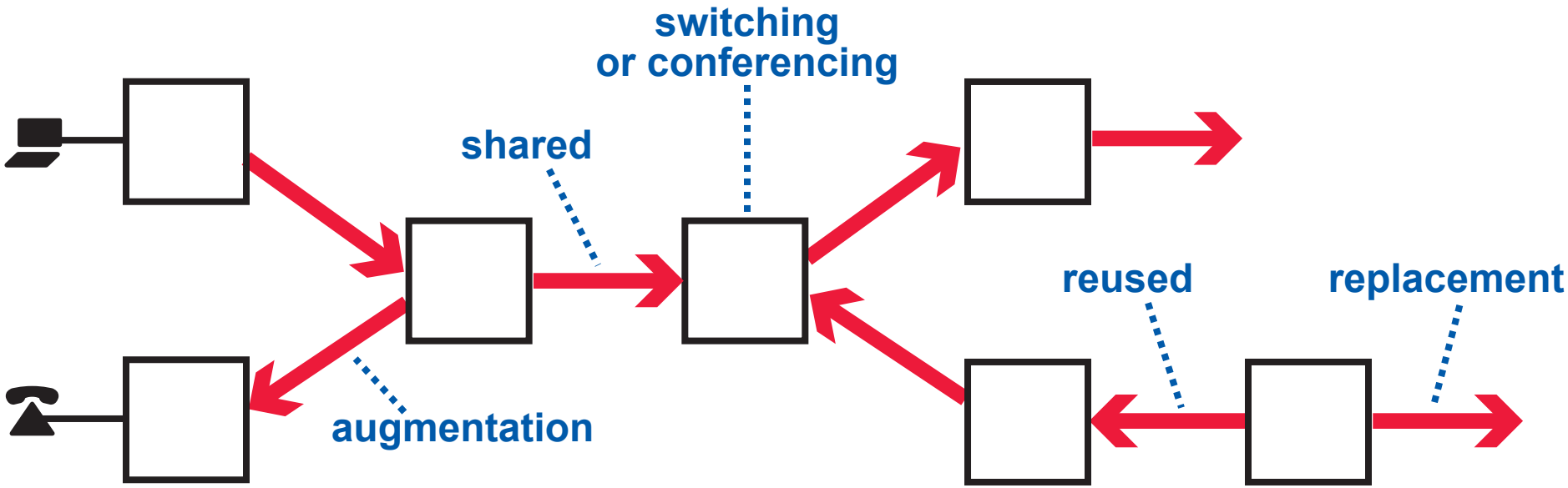
DESPITE THE EXCEPTIONS, IDEAL ADDRESS TRANSLATION HAS PROVEN VERY USEFUL BECAUSE . . .

. . . even a subnetwork can have very complex feature interactions

. . . principles, constraints, properties, and reasoning are all models that we approximate as closely as possible

. . . it helps us understand infrastructure requirements

INSIGHT ACCELERATES INSIGHT



THIS IS PART OF A DFC USAGE—NOW IT SEEMS POSSIBLE TO ANALYZE THIS!

including:

- **extend ideal address translation to unrestricted usages like this one**
- **strengthen the properties, because the model describes more of what is going on**

e.g., prove that the current far-party address correctly identifies who you are talking to

(before, the model only told you about how the usage was constructed by routing)

CONCLUSIONS

TELECOMMUNICATION REQUIREMENTS USED TO SEEM INTRACTABLE, AND NOW THERE IS A FEELING OF REAL PROGRESS

OBSERVATIONS ABOUT WHAT WORKS

- sometimes architecture must precede requirements
- I made most of this progress after I stopped trying to accommodate all legacy systems
- above all, be domain-specific

NOW:

http://www.research.att.com/info/pamela

AFTER 15 June 2003, including references on address translation:

http://www.research.att.com/projects/dfc