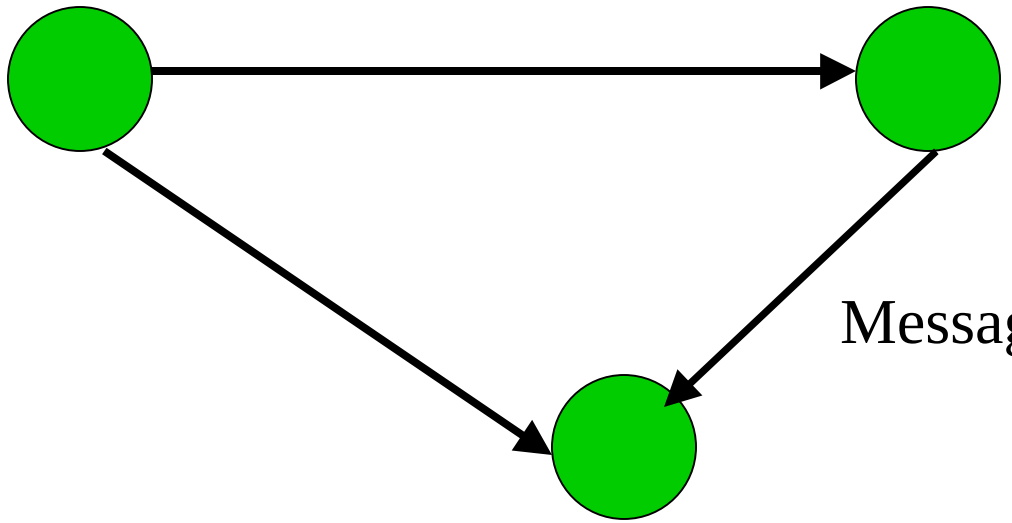
Chapter 2

Design Issues

Part 2.2.

the process-message graph abstraction



- works with active objects too

Performance

- achieving it by a near-optimal match of resources to workload
- (NP-hard problem, no doubt)

Consistency

- maintaining it at affordable cost . . .

Details:
Naming

- we *resolve* or *translate* the name

  (object identifier; who it is)

  e.g. eric manning

- into an *address*

  (where it is; how to get to it)

  e.g. 604 721 0976

  (BC area; 721 exchange; line 0976)

NOTE:

- distinction between *address* (how to get there; how to access the object) and *communication identifier*:
- all addresses are com idents; converse false.
- com ident concept: address resolution often multistage; each comident passed to the next lower level for further resolution,
- final comident must be an address.

- e.g. internet IP address 192. 135. 231.4
  - is a concatenation of net and subnet and host *names;* thus not an address.
  - no location or routing information.
  - further translation required (in the subnet).

How to design naming systems;

design

1] name space and structure

- (infinite? or finite? how big?)
- (structured? flat?)

2] resolution machinery:

- usually a service ("name service")
- sometimes merged with resource named,
  for efficiency
  - (e.g. filename -> disc address translation)
- parts of the *translation table* often cached locally
  for efficiency, but consistency problems arise ...

- provision of *context* for the name:
  - specify the namespace in which it is considered to live

Example:

- **604 721 0976:**

- 604 belongs to the namespace of area codes

- 721 belongs to the namespace of office codes *within* areacode 604.

- 604 721 0976
  - not actually an address
  - says where the phone is (partial address: Feltham Road in Victoria in BC)
  - but not which line it's attached to (wireframe address)
  - and nothing about how to get there (route)

- 604 721 0976

- a *partial translation*

  from ericmanning

  to the address of his home phone

Example:

- /etc/passwd
  - etc is in the context "root"
  - passwd is in the context (under the directory) "etc"

Making namespaces unbounded:

- 1. use hierarchies, or
- 2. allow namestrings to grow in length without bound

Protection:

- make identifiers hard to fabricate

- naming service checks rights of clients to hold identifiers before providing them (capabilities)

Communication
Purposes

- 1. data transfer

- 2. synchronization of sender & receiver
  (at some T, S was in s_1 and R was in s_2 .)

degrees of synch:

- blocking send, blocking receive
  - common, tightest sync, least concurrency
- blocking send, nonblocking receive

  and
- nonblocking send, blocking receive
  - intermediate in both

Degrees of synch:

- nonblocking send, nonblocking receive
  - Loosest synch; most concurrency

Additional structure above message sending:

- remote procedure call
- client-server
- multicast