JavaOne℠

Sun's Worldwide Java Developer Conference

JavaOne℠

Sun's Worldwide Java Developer Conference

Multiuser Environment

Social-oriented Online System: Requirements, Examples, Futures

Douglas Crockford

Electric Communities

Cupertino, California

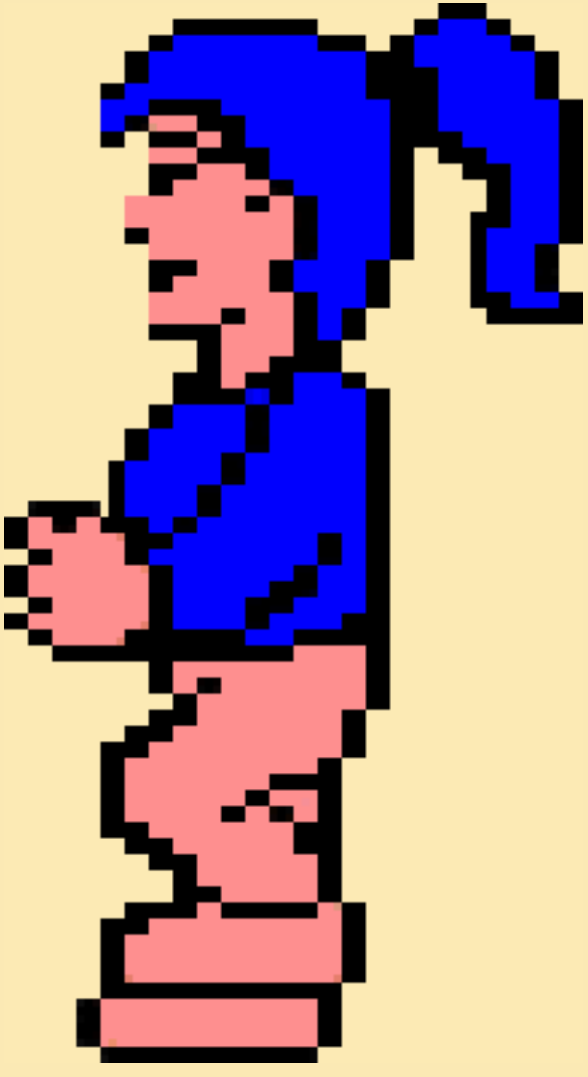
Java™-based Names

- Java™ and other Java™-based names and logos are trademarks of Sun Microsystems, Inc., and refer to Sun's family of Java™-branded products and services.

Electric Communities

Lucasfilm's Habitat (1985)

- First graphic, online community
- Avatars
- Commodore 64
  - 64K RAM
  - 300 baud modem
  - 1MHz 6502
  - Slow floppy disk



Lucasfilm's Habitat

- Habitat provided an acceptable level of social interaction for tens of thousands of members
- How did we do it?
- Object oriented protocols

Lucasfilm's Habitat

- Habitats are excellent places to learn about the sociology of online communities
- You can't dictate human behavior
- Systems must adapt to evolving social context

AMiX
American Information Exchange

- Information marketplace
- Buyers and sellers of information
- Static and dynamic information

AMiX AMERICAN INFORMATION EXCHANGE

AMiX Electronic Commerce

- Order entry & credit card payment
- Contracting, dispute resolution, reputation system
- Social environment
- Mediation services

WorldsAway

- Designed for Fujitsu
- Based on the original 1985 design
- Available through CompuServe

Experience

- Only three graphic virtual communities have generated revenue on an ongoing basis
- We designed all three
- Human-centered design

Social Systems

- Recreation, commerce, collaboration, education, support
- 60’s: Timesharing
  - Chat
  - Games
  - Document exchange
- Engelbart at SRI

Architecture for Social Networking

- 70’s: mainframes and minis
- 80’s: standalone PCs
  - The dark ages of social computing
  - X-Modem
  - Email eclipsed by fax
- 90’s: networked PCs

The Internet

- The greatest value of the Internet was and always will be a medium in which individuals could interact with each other for their mutual benefit.

People in Cyberspace

- The challenge is to give people a reason to go there
- Content is just one of the currencies on the Net
- The most important currency is the relationships between people

Rights

- Right to assemble and speak
- Right to protect individual privacy
- Right to trade with anyone
- Right to meet our neighbors
- Right to meet people in other places
- Without the danger of viruses, trojan horses, and other security threats

Scalable

- Scalable social systems are hard to make
- Sociological issues
- Technological issues
  - The Java™-based platform is great as far as it goes, but lacks features needed for scalable, distributed systems

Distributed Programming

- Remote procedure call
- Remote message invocation
- Not well-suited to social mediation

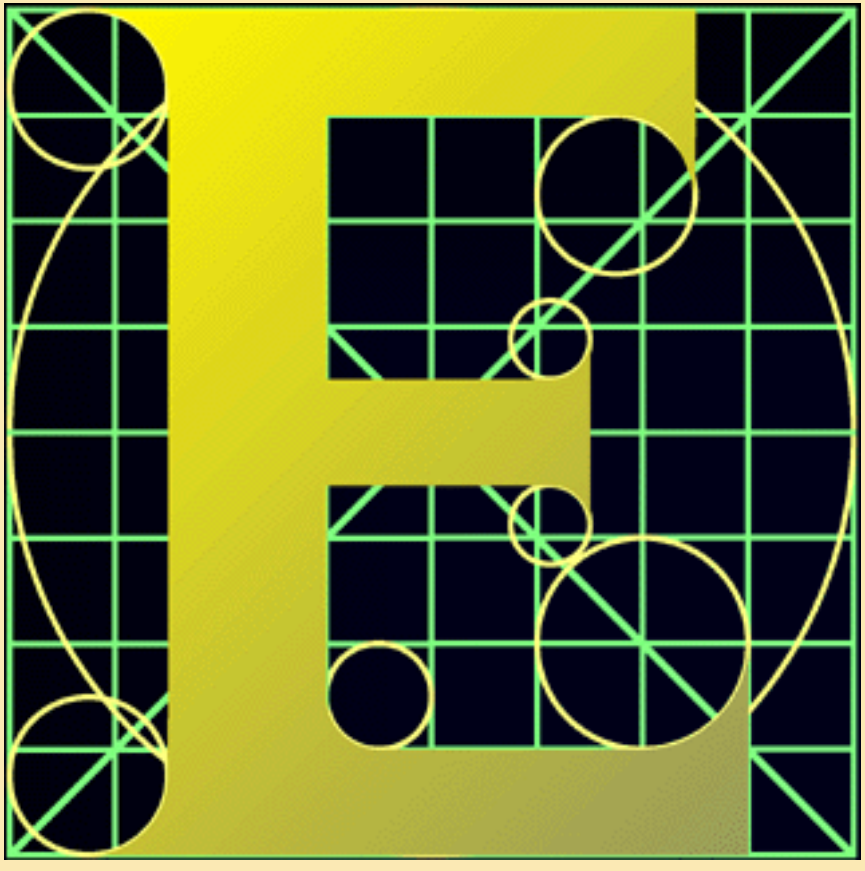
Social Oriented Distributed Programming

- Interaction between people
- Social environments have a much greater emphasis on communication
- Dynamic

Four Requirements

- Communications
- Concurrency
- Security
- Optimistic computation

The E Extensions



E is Java™- powered

- The best of two powerful programming paradigms
- E is implemented as a compiler, runtime, and class libraries
- It was not possible to add this level functionality by simply adding classes

E: Communications

- The EObject
  - No public methods
  - Message passing
    - Real messages, not polymorphic subroutine calls
    - One way
    - Immediate, asynchronous, optimistic, non-blocking

`EObject <- message_name (parameters);`

- Messages can be sent to EObjects on the same machine or across the network
- You can test eclasses in a single machine, and then use them in a networked configuration with no recoding or recompilation

Message Passing

- Message passing is a natural way to work with networks
- The programming model matches the communications model

Automatic Connection Management

- No sockets
- No streams
- No threads
- No low-level protocols
- Just EObjects and messages

References

- If you have a reference to an EObject, you can send a message to it
- Sources of references
  - Initialization
  - Creation of EObjects
  - Messages
- References cannot be forged: *Capabilities*

Virtual Networks

- By propagating EObjects, it is easy to build and maintain virtual networks
- Foundation for communities

Distributed Garbage Collection

- Keeps EObjects alive that are referenced only from the network
- Reclaims those EObjects when they become unreferenced
- Reclaims unreferenced distributed cyclical structures

E: Concurrency

- EObjects are active objects
- High levels of concurrency without threads and synchronization
- No blocking or suspension
- Mutual exclusion can be assured by encapsulating critical data in EObjects

Processing Loop For an EObject:

- Receive message
- Execute Emethod, which may
  - Change internal state
  - Create objects
  - Send messages to EObjects
  - Call methods of Java™ objects
- Repeat

Deadlock Avoidance

- The worst kinds of bugs
- Realtime interactions between objects, possibly affected by interactions with other machines
- These failures can be intermittent, often unreproducible, and are extremely difficult to debug

E: Security

- Meaningful relationships depend on trust
- Commerce depends on trust
- Secure systems really are necessary
- We need a better balance between power and safety

Trust Management

- The E Trust Manager provides for the signing of classes and packages, providing a tamper-proof seal identifying the source of the software and proof that it has not been altered or extended

Trust Management

- The central question that the E Trust Manager is concerned with is not "Can this class be trusted?," but "What can this class be trusted *with*?"
- E supports the creation of execution environments which are highly dynamic and highly restrictive in the interfaces provided to alien software

Trust Management

- Under the E Trust Manager, classes will be loaded only if they are trusted to use the classes they require
- We do this by issuing “class capabilities” relying on positive assertions of trust

Trust Management

- Claims of the form that “Class A can be trusted with Class B” can be made by the maker of Class A or Class B, but that claim is recognized only if the source of the claim is trusted
- The scheme is fine-grained, flexible, extensible, and delegatable

Capability Semantics

- Object
- Class
- Message

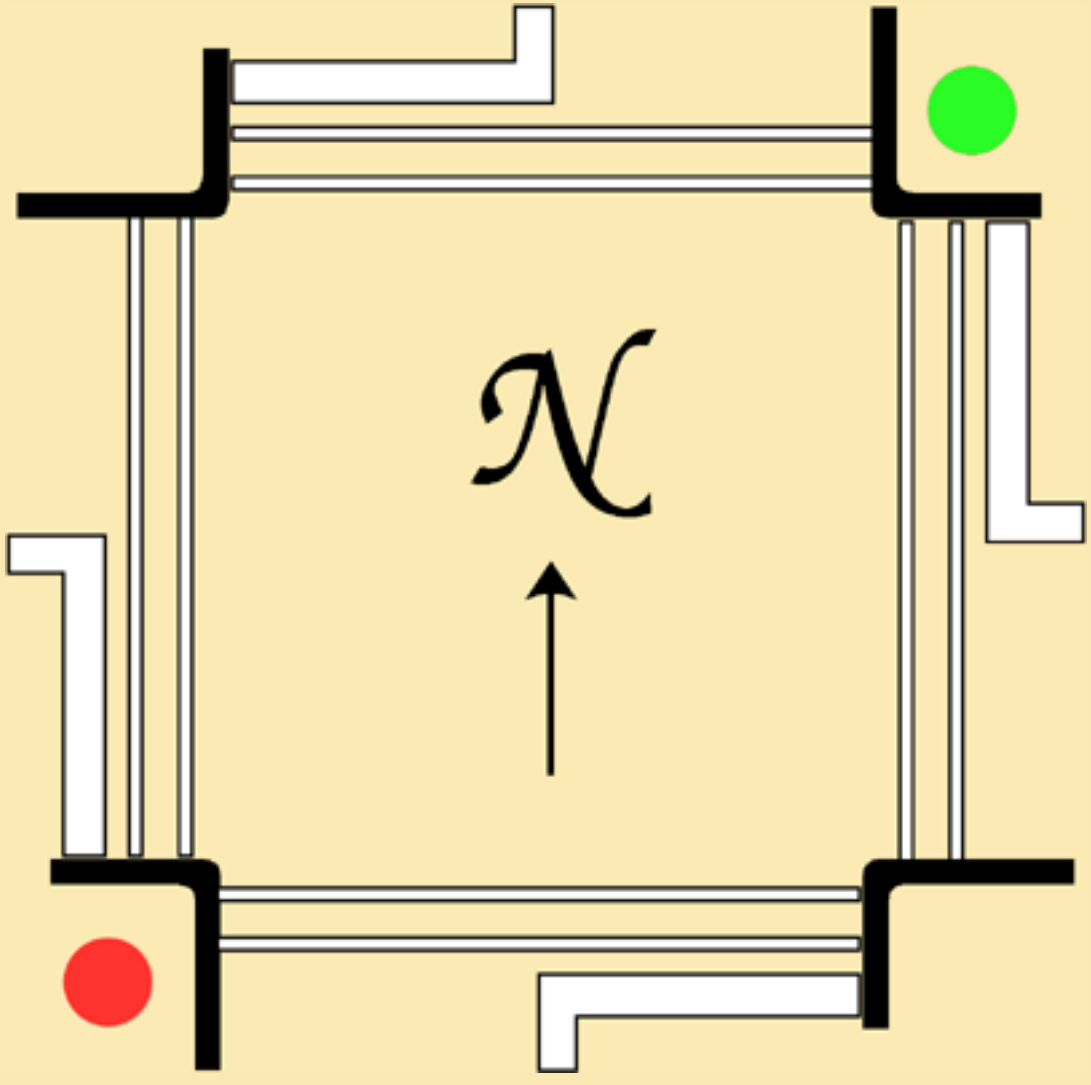
E: Optimistic Computation

- Inherent delays in the network
- When people wait, they get irritable
- Latency is not going to get better
- Optimistic Computation is a tool for dealing with latency

Optimistic Computation

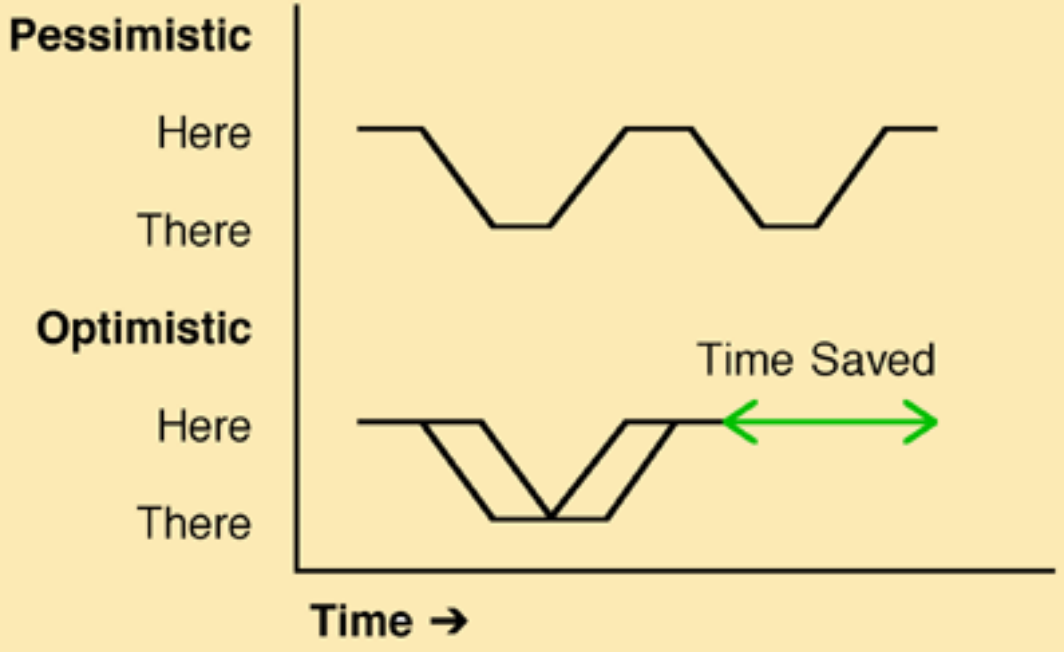
- Assume that everything is going to work and keep going
- This tends to increase parallelism, balance loads, and reduce cumulative latency
- Cumulative latency is a major killer of distributed applications

Flexible Sequencing



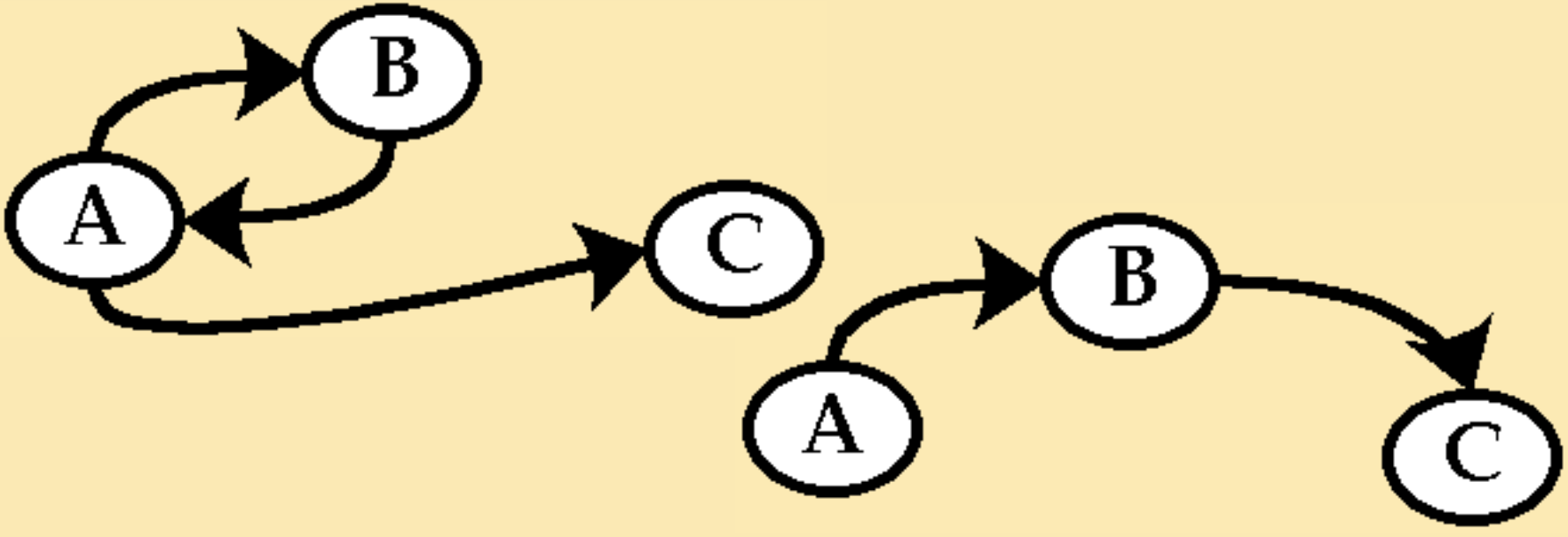
Latency Compensation

- Reduce cumulative time spend waiting for network-based delivery



Communication Avoidance

- Dynamic routing
- The most effective way of making the network seem faster is to use it less



Channels

- A special object which can receive messages and then forward them on
- The sender does not need to know where the message ultimately will go
- The channel can begin receiving and collecting messages before it is given the forwarding address

Channels

- Every channel has a component object called a distributor
  - `distributor <- forward (reference);`
- Channels can simplify program design by reducing the need and complexity of message naming

Channels Can Reduce Latency

- Futures
- In many cases, computation involving time-distant results can proceed even if the results are not yet known
- Messages can be sent to EObjects before they are created
- References not-yet-created to EObjects can be sent

Trading Table Demonstration

- Metaphor for a more general model of net commerce
- Objects have a sense of ownership
- Ownership can be transferred using the trading machine object

Trading Table Demonstration

- Simple virtual space shared by three computers
- There is no central server
- Pretty easy to write using E
- Available at
  - `http://www.communities.com/`

E

- Electric Communities developed E for its own use
- We wish to share it with you
- E will be used as the implementation language for global marketplace

E Is Available Now

- http://www.communities.com/

You are Such a Lovely Audience

- We’d like to take you home with us
- We’d love to take you home

The End