P2P-Usenet
Charter

Project Statement

We aim to create a system where one peer can message another peer, and other peers within the network can attach, monitor, and participate in that "message/dialog" to achieve dynamic multimedia and information exchange.

Business Drivers/Background

USENET is a decentralized, client-server discussion system that is based upon a general Unix-to-Unix-Copy architecture (wikipedia). Users read and post email-like messages to categories known as "Newsgroups" (wikipedia). Current implementations have clients post articles to local servers, which in turn exchange the messages between each other. Naming conventions/spaces have been set to allow logical grouping of messages/ newsgroups. Every server that subscribes, hosts, or shares a newsgroup/namespace holds exactly one message on the machine. This allows clients to go back and search through old messages as well as pull down new messages sent the the newsgroup. Doing this allows usenet to achieve data redundancy and storage

Vision

We would like to move this system even further decentralized by removing the client-server architecture and replacing it with a P2P architecture. Namespaces are replaced with the ability of peers to "tag" messages. This lets the users of network decide which information is the most important and to identify it with the most common and popular name. Messages are copied to a peer's computer when they send, receive or subscribe. Metadata associates with the message is also bundled. This includes tags and an index of keywords from the message. When the message is not text based, only tags are used. Once all peers in the network delete all their messages from a given dialog, that dialog is said to have "expired" and no longer is relevant information for anyone. In this way, data storage and redundancy is decided by the dynamics of the peer-to-peer network itself. When a user searches for a message, the search crawls across the messages on peers' machines, inspecting the metadata associated with messages (tags and indexed keywords) and reports back the results. Only subscribers can tag messages. This is a form of quality assurance, that only those interested in the material will know how to appropriately label the material. Users can subscribe to tags, where every message that is given a specific tag is sent to the users who have subscribed to that tag. Users can also subscribe to a dialog that is happening between two people, where peer ids are used to determine the publishing destination of the article/message.

Goal Statement

Our goal is to create a system that allows dynamic multimedia and information exchange on a P2P network that mimics the behavior of the current client-server usenet system.

Objectives

Specific, measurable, attainable, realistic, time-bound, concrete objectives for achieving the goal:

We aim to create a scalable decentralized P2P network that allows

- one peer to send another peer a message
- outside peers to read that message
- outside peers to have the dialog between peers also be sent the them, like being a CC on an email
- outside peers to tag the dialog with keywords
- outside peers to search for dialogs that exist in the network (search will span across content and tags shared by all peers on the network)
- outside peers to browse dialogs based on how they were tagged
- Peers to delete dialogs they have sent or received
- The ability to send not just text, but various types of multimedia as a dialog

Scope

What will and will not be delivered as part of this project.

In Scope:

- P2P messaging system supporting text and multimedia sharing
- Tag-based search functionality across P2P network

Out of Scope:

- Plugin system for the P2P network (unless a third party P2P framework offers it)
- Media players or viewers associated with the data shared by the application
- Cross-platform functionality
- Data visualization or history based tracking

Schedule

Project start and end dates, milestones and/or phases

Project start date: October 23, 2007

Phase I

Please see the "Roadmap" on our Trac

Milestone- Application Description Document is completed and accepted by the professor
Milestone- The architecture design document and supporting documents are completed and approved by the team
Milestone- The preliminary report and presentation are completed and accepted by the professor

Phase II

Milestone- The system is implemented and the team confirms it meets our specifications
Milestone- The final report is completed and accepted by the professor
Milestone- The final presentation is completed and accepted by the professor

Project end date: December 07, 2007

For specific Dates and Progress, please see the Roadmap on our Trac

High Level Deliverables

The tangible, verifiable outcomes of work that satisfy the project objectives

Deliverable	Description
Application Definition Document	2 Page overview (Charter and Scope)
Design Document and Architecture	Architectural plan and support documents
Preliminary Report	Review of our architecture (template supplied)
Preliminary Presentation	Presentation summarizing the report
Implementation	A running system of the architecture supplied
Test Suite	Proof that our system is validated and verified against requirements and our model
Final Report	Analysis of our system (template supplied)
Final Presentation	Presentation summarizing the report

Project Organization

Roles & Responsibilities

Roles and responsibilities assigned to each individual or group resource to the project.

Project Sponsor: Dr. Peppo Valetto

- Supply final grades on deliverables
- Provide guidance
- Evaluate deliverables prior to due date if requested
- Define deliverables

Project Sponsor: Sunny

- Aid Peppo in all tasks

Project Member: Paul deGrandis (pd442)

- Software Design, Management, Documentation, Implementation, and Testing
- *"I will always make sure the team is staying on track"*

Project Member: Brian Vass (bmv82)

- Software Design, Management, Documentation, Implementation, and Testing
- *"I will ensure that our project scope meets and does not exceed our original vision"*

Project Member: Craig Eisenberger (cae24)

- Software Design, Management, Documentation, Implementation, and Testing
- *"I will assure documentation is accurate and consistent throughout the project"*

Project Member: Matt Oyer

- Software Design, Management, Documentation, Implementation, and Testing

Team Composition

- Paul deGrandis
- Brian Vass
- Craig Eisenberger
- Matt Oyer

Approach

The team will work distributedly using the Internet and related technologies to achieve milestones, complete deliverables and finish the project. Trac, an integrated SCM and Web-based project management tool will be used to issue tasks and bugs, track the completion of milestones, browse source code and documents, and use to create documents and documentation via the integrated wiki. Subversion will be used as a source code and document repository (SCM). The team will communicate via email, instant message and document creation via the wiki. We will work together to build the architecture, research possible solutions, examine similar implementations and study the positive and negative aspects of design decisions the team makes while we complete our list of objectives and construct our deliverables.

Assumptions and Constraints

Assumptions:

The events that need to occur for the project to be successful but are outside the total control of the team

- We assume that there will be no requests for functionality outside of our scope at anytime during development
- We assume that there are open source or free software solutions we can leverage to achieve our goal
- We assume we can treat certain modules of code as a blackbox, given the above assumption
- We assume that at no point will the project deliverables change and that we can plan on a steady schedule

Constraints:

Limitations generally outside the control of the project which may negatively impact the project scope

- We are limited on time and have a fixed schedule which cannot be expanded
- Due dates are hard fixed and must be met
- The team cannot grow
- Work will be decentralized and not full time by all members
- If we use a third party solution for any module, we are limited by that module's capabilities

Performance Measures/Outcomes

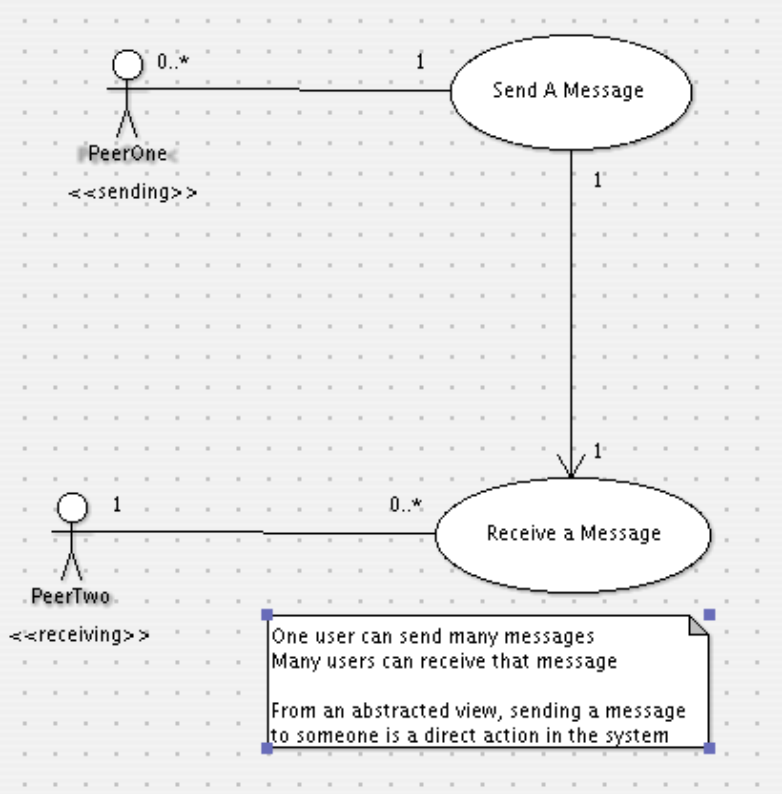
Our schedule is fixed and our deliverables are evaluated and graded by Peppo and Sunny. We will use Trac to track our progress on issues, tasks, and bugs associated with each milestone and deliverable. The team will have an informal post-mortem after each deliverable via email and the wiki to discuss what was done correctly and what we need to improve upon for the next deliverable. We will utilize each individual's skillset to achieve all goals for the system and the class. We may utilize fast-tracking to better manage the fixed deadlines and the size of tasks needed to meet deliverables and milestones.

Acceptance

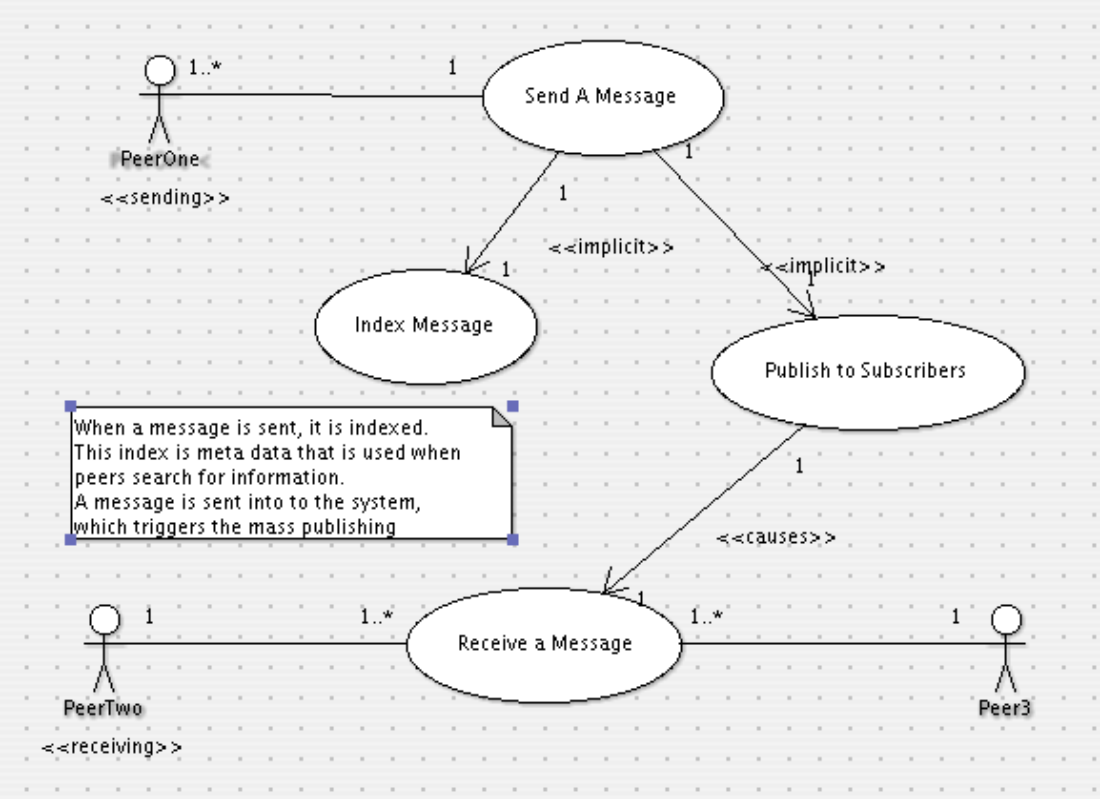
Unanimous approval by Key Stakeholders*(Sunny/Valetto)* for the final charter document.

Diagrams

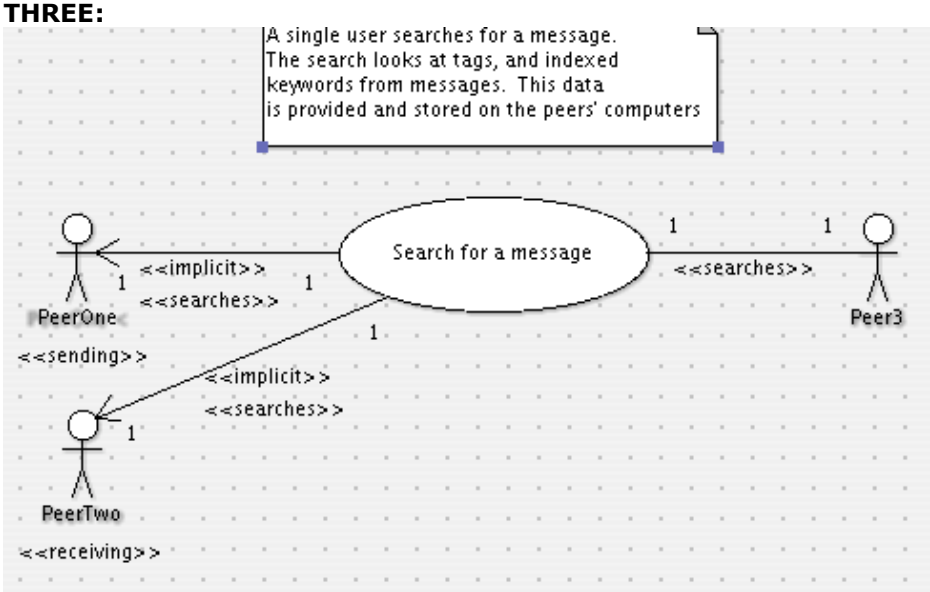
Below are some simple use-case diagrams to illustrate our vision further.
ONE:



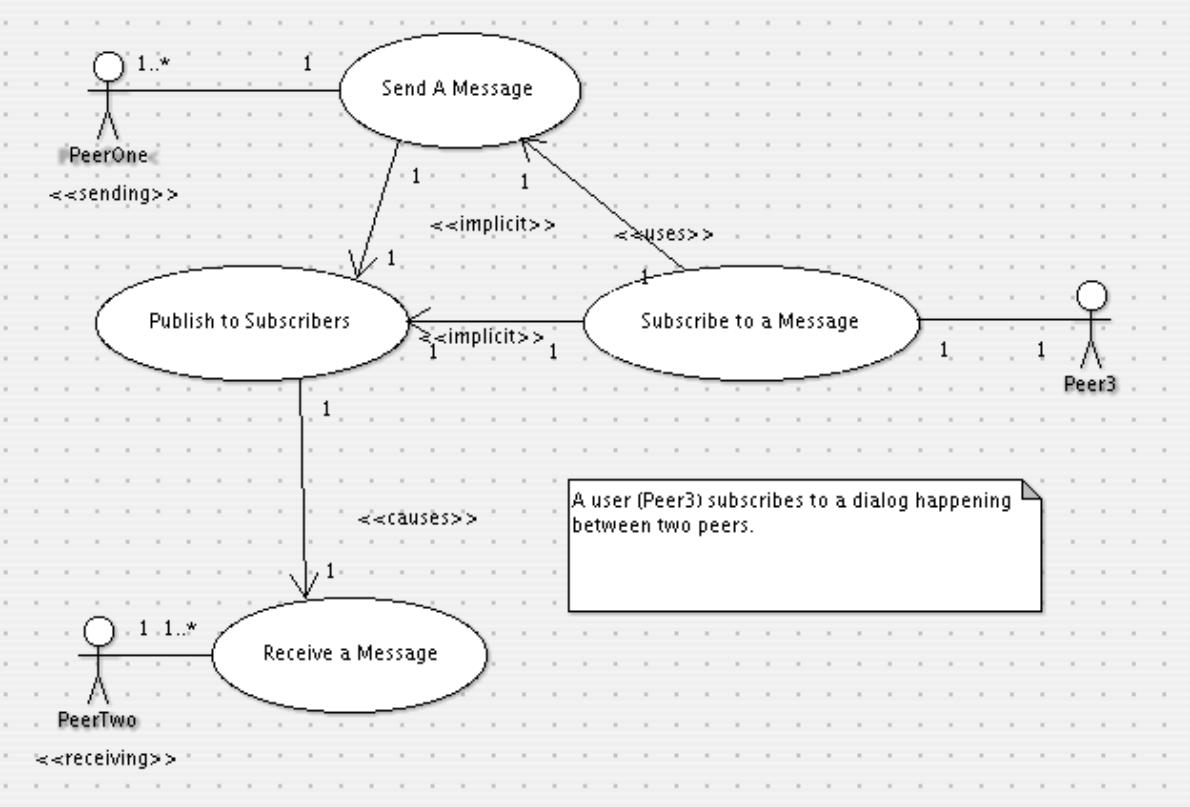
TWO:


1..*
1
Send A Message
PeerOne
<<sending>>
1
1
<<implicit>>
1
<<implicit>>
1
Index Message
Publish to Subscribers
When a message is sent, it is indexed.
This index is meta data that is used when
peers search for information.
A message is sent into to the system,
which triggers the mass publishing
1
<<causes>>
1
1
1..*
Receive a Message
1..*
1
PeerTwo
<<receiving>>
Peer3

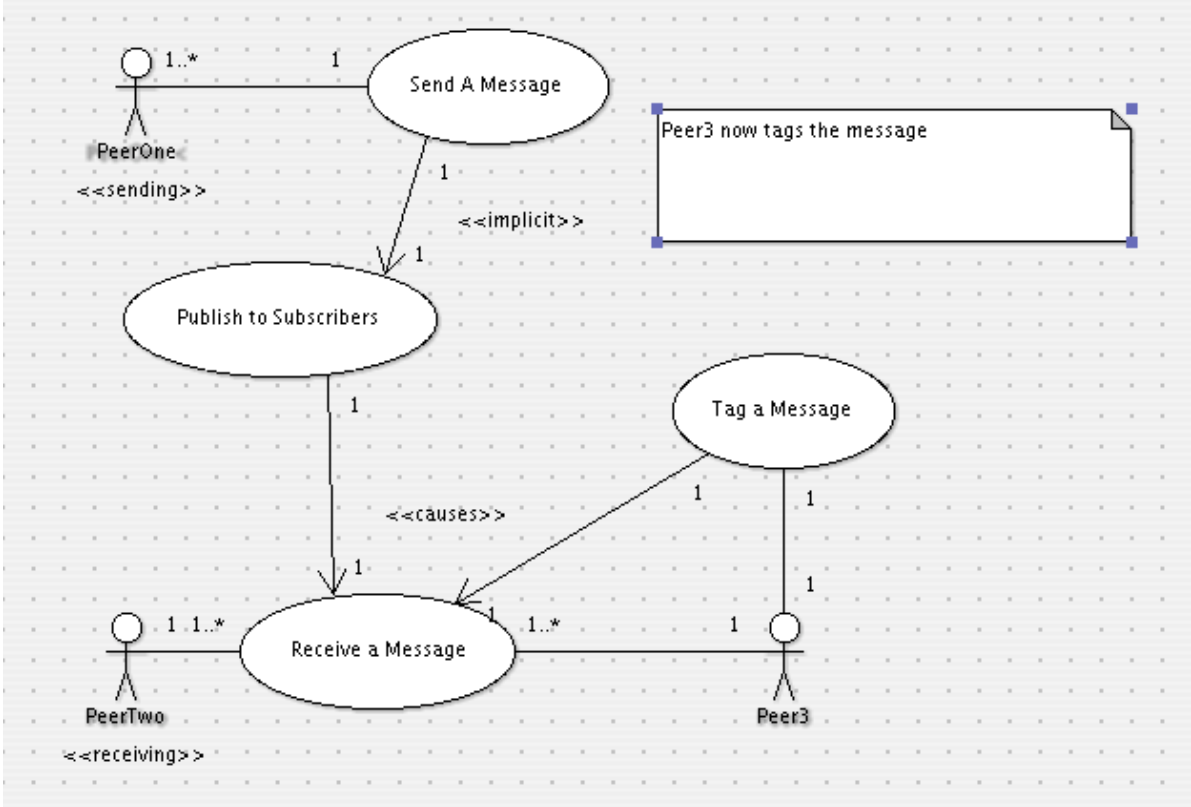
THREE:


A single user searches for a message.
The search looks at tags, and indexed
keywords from messages. This data
is provided and stored on the peers' computers
Search for a message
1
1
<<searches>>
Peer3
1
<<implicit>>
1
<<searches>>
PeerOne
<<sending>>
1
<<implicit>>
<<searches>>
1
PeerTwo
<<receiving>>

FOUR:


1..*
1
Send A Message
PeerOne
<<sending>>
1
1
<<implicit>>
<<uses>>
1
1
Publish to Subscribers
<<implicit>>
Subscribe to a Message
1
1
1
1
Peer3
1
<<causes>>
A user (Peer3) subscribes to a dialog happening between two peers.
1
1
1..*
Receive a Message
PeerTwo
<<receiving>>

FIVE:


1..*
1
Send A Message
PeerOne
<<sending>>
Peer3 now tags the message
1
<<implicit>>
1
Publish to Subscribers
Tag a Message
1
1
1
<<causes>>
1
1
1
1
1..*
Receive a Message
1..*
1
PeerTwo
Peer3
<<receiving>>

We, the undersigned project members, have reviewed this document and approve its contents:

Name and Title Signature Date

Attachments

- P2PUsenetUsecase1.png (14.4 kB) - added by se310 on 10/23/07 05:32:58.
- P2PUsenetUsecase2.png (20.0 kB) - added by se310 on 10/23/07 05:33:07.
- P2PUsenetUsecase3.png (14.2 kB) - added by se310 on 10/23/07 05:33:18.
- P2PUsenetUsecase4.png (19.5 kB) - added by se310 on 10/23/07 05:33:30.
- P2PUsenetUsecase5.png (18.6 kB) - added by se310 on 10/23/07 05:33:42.