A Multi-Agent implementation of Social Networks

Enrico Franchi (e.franchi@ce.unipr.it)
A Social Network (SN) is a connected graph of public and/or semi-public profiles.

A Social Network System (SNS) is a software system that supports the persistent storage of SNs and that provides means to update, to add and to query information.

We also expect a SNS to suggest proactively possible acquaintanceships among users, using the information in user profiles.
Features of Existing SNSs

- Lots of different social network systems exist
- They are centralized systems
  - Relatively easy to suggest contacts
  - Privacy concerns with user data
  - Funded through ads (*stronger privacy concerns*)
- The goal is to build a Distributed SNS (DSNS) using Agents
Small World Phenomenon

- Milgram original experiment with snail mail, 1967 [1]:
  - 6 degrees of separation
  - “Searchability” of social networks
- Confirmed by more recent experiments
- Social networks are “small-world” networks (Strogatz and Watts [2])
  - Neural network, the power grid and the collaboration graph of film actors
- Searchable small-world networks have “power-distribution” for the “random” short paths [3]
Small World Phenomenon

- Milgram original experiment with snail mail, 1967 [1]:
  - 6 degrees of separation
  - “Searchability” of social networks
- Confirmed by more recent experiments
- Social networks are “small-world” networks (Strogatz and Watts [2]):
  - Neural network, the power grid and the collaboration graph of film actors
- Searchable small-world networks have “power-distribution” for the “random” short paths [3]
ReferralWeb

- Multi-agent platform to “search” for experts in a user’s social network
- Exploits “searchability” of social networks
- Completely different from modern Social Networking Systems
Foner, 1997 [5]

“Matchmaker” system to bring together people with similar interests but *mainly* intended to “find experts”

Different from “modern” SNSs where users want to connect to friends
Mika, 2005 [6]

Extraction, aggregation and visualization of online social networks

Semantic technology for reasoning with personal information extracted from a number of electronic information sources

Used for social network analysis and for generating a web-based presentation
Matsuo et. al, 2007 [7]

Several advanced techniques to extract relations of persons, to detect groups of persons, and to obtain keywords for a person

Uses multiple sources for the data (among them, FOAF profiles)
DSNS Architecture

- Each user is represented by a software agent
  - The agent mediates connections to user data
  - The agent proactively searches new contacts
- Each agent has a unique identifier (user nick, email, ...)
- Identifiers should be resolved to agents
- Software agents communicate through messages
- Public/private keys are used to sign and encrypt communication
Agents should build their user’s social network

Agents shall only disclose the minimum amount of information

Agents can communicate only with other agents they “know”, i.e., they are connected with
User profiles

- Data come from different sources (posts, queries, tags, profiles)
- *Non restrictive assumption*: all the relevant information is in a RDF profile (FOAF)
  - the owner is put in relation with other entities
  - we “extract” these relationships and derive possible acquaintances

![Diagram showing relationships between user profiles and degrees](image-url)
User profiles

- Data come from different sources (posts, queries, tags, profiles)
- *Non restrictive assumption*: all the relevant information is in a RDF profile (FOAF)
  - the owner is put in relation with other entities
  - we “extract” these relationships and derive possible acquaintances

```
P1 <<knows>> P2
```

Attended course in CE in Parma with
Example

<foaf:Person>
<foaf:name>P1</foaf:name>
<doac:education>
<doac:Degree>
<doac:title>Computer Engineer</doac:title>
<doac:organization>University of Parma</doac:organization>
</doac:Degree>
</doac:education>
</foaf:Person>

<foaf:Person>
<foaf:name>P2</foaf:name>
<doac:education>
<doac:Degree>
<doac:title>Computer Engineer</doac:title>
<doac:organization>University of Parma</doac:organization>
</doac:Degree>
</doac:education>
</foaf:Person>
Connection Discovery Algorithm

A : Agent

1: FindConnection(L, ex)

B : Agent

1.1: RequestConnection(A, L)

2: AcceptConnection(A, L)

C : Agent

[if C "likes" A]

2.1: AcceptedConnection(C, L)

2.2: AcceptedConnection(A, L)

[else]

3: RefuseConnection(A, L)

red arrows if information transmission occurs
Implementation details

- Use HDS as the basis
- Reasons:
  - Algorithms are easily conceivable as typed messages exchange
  - Performance may be important
  - Agents may reside on heterogeneous nodes
## Implementation with HDS

1 Agent ↔ 4 Processes [2 Actors, 2 Servers]

<table>
<thead>
<tr>
<th>Process</th>
<th>Type</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proc1</td>
<td>Agent</td>
<td>Searches new connections according to available data</td>
</tr>
<tr>
<td>Proc2</td>
<td>Agent</td>
<td>Brokers connections with possible mutual friends</td>
</tr>
<tr>
<td>Proc3</td>
<td>Server</td>
<td>Accepts/refuses connections proposed by Proc1 and Proc2</td>
</tr>
<tr>
<td>Proc4</td>
<td>Server</td>
<td>Mediates access to data</td>
</tr>
</tbody>
</table>
Future work

- Experimental study on the algorithm
- Devise mathematical models based on first item
- Develop distributed variants of algorithms trying to “match” FOAF profiles
- Add new data sources (posts, image tags)
- Explore engineering issues (e.g., what happens when the user turns off the machine)
Thanks for your kind attention!

A Multi-Agent implementation of Social Networks

Enrico Franchi (e.franchi@ce.unipr.it)
References


