

# AMERICAN SOCIOLOGICAL REVIEW

OFFICIAL JOURNAL OF THE AMERICAN SOCIOLOGICAL ASSOCIATION

---

**ONLINE SUPPLEMENT  
to article in**

AMERICAN SOCIOLOGICAL REVIEW, 2004, VOL. 70 (OCTOBER:823–842)

## **SOCIAL STATUS IN AN OPEN-SOURCE COMMUNITY**

Daniel Stewart  
*Washington State University*

**SECTION A.  
SAMPLE PERSONAL WEB PAGE FROM  
ADVOGATO.ORG**

(Source: <http://www.advogato.org/person/zhaoway/>)

Personal info for zhaoway

This person is currently certified at Journeyer level.

Name: zhaoway

Homepage: <http://www.zhaoway.com/>

This person is:

- a Developer on project Debian.
- a Documenter on project Evangelism.

Recent diary entries for zhaoway:

**14 Apr 2002 »**

Gradually I'm having a different attitude towards correctness proving. But I need to read more. And I need to code more. I heard Erlang is a good real time language and is functional. But I will be far away from mastering it that I would be able to put it into work. Generally, though I feel functional programming is difficult, Clean makes me feel that it is interesting, and could be easier later on, at least I hope so. So I'd like to do more in that. And in the pure way.

Uh, the Object I/O library of Clean, which is difficult. I think the difficulty is mainly because of the type system and the pure nature of Clean. Whatever, it is difficult, albeit the *uniqueness* discipline is pretty simple and straight. I spent the early morning solving a small problem of local state passing, and finally did it. Hee hee.

From Scheme I learnt *continuation* which is by far the most *cool* thing I learnt from programming languages, but I have to say that impure functional languages gives little in helping learning the functional style.

I'm more or less lost in knowing what the hell the benefits of macros are. That means programming is a difficult topic, I mean, only after I do a project which materialize the power of macros, can I really understand the power of macros. My beliefs in macros would be just lies 'till that day comes.

**Lone Wolf**

I read that Hermann Weyl being called the *lone wolf* among Hilbert's students. It's amazing that an *egg head* mathematician being called *lone wolf*. Previously I thought only characters in Hemmingway or Jack London's books or from Schwazeneiger's movies will

ever be called *lone wolf*. Funny. You'd love mathematicians.

**Elephant**

Okay, I think I could name my warm-up project in the Clean language now. The name is *elephant*. It's a program playing Chinese Chess a la Xiangqi through TCP. Only 10k characters have been written so far. But I did draw all of the 14 pretty pictures of the chessmen using only the mouse and successfully loaded them through the API jungle of Clean's Object I/O library. I am proud of it. Hehehe.

65 older entries...

This person has certified others as follows:

- zhaoway certified foka as Journeyer
- zhaoway certified spacehunt as Journeyer
- zhaoway certified perlamer as Journeyer
- zhaoway certified wichert as Master
- zhaoway certified tausq as Journeyer
- zhaoway certified evo as Journeyer
- zhaoway certified andersee as Master
- zhaoway certified claviola as Journeyer
- zhaoway certified sye as Apprentice
- zhaoway certified bratsche as Journeyer
- zhaoway certified atai as Journeyer
- zhaoway certified ianmacd as Journeyer
- zhaoway certified fxn as Journeyer
- zhaoway certified thom as Journeyer
- zhaoway certified forrest as Journeyer
- zhaoway certified crhodes as Journeyer

Others have certified this person as follows:

- foka certified zhaoway as Apprentice
- spacehunt certified zhaoway as Journeyer
- perlamer certified zhaoway as Journeyer
- jrf certified zhaoway as Journeyer
- fxn certified zhaoway as Journeyer
- sye certified zhaoway as Journeyer
- evo certified zhaoway as Journeyer
- Joy certified zhaoway as Apprentice
- chalst certified zhaoway as Journeyer
- forrest certified zhaoway as Journeyer

Certify zhaoway as:

See the Certification overview for more information.

[ [Home](#) | [Articles](#) | [Account](#) | [People](#) | [Projects](#) ]

**SECTION B.**  
**ADVOGATO CERTIFICATION SYSTEM**  
(Source: <http://www.advogato.org/certs.html>)

**MASTER**

A Master is the principal author or hard-working co-author of an "important" free software project, i.e. one that many people depend on, or one that stands out in quality. A Master has command of the tools and is an excellent programmer. Generally, a Master works equivalent to full time (or more) on free software. Ideally, a Master writes clearly about the work and its broader context, and serves as a mentor to others in the free software community.

**JOURNEYER**

Journeyers are the people who make free software happen. A journeyer contributes significantly to an important free software project, or is the author of a useful or technically innovative project. A Journeyer is generally a competent programmer, but significant contributions of documentation, artwork, or other non-code goodies counts too. Ideally, a Journeyer works with others in the free software community to polish and refine the library of free software. While not necessarily the equivalent of full time, a Journeyer spends a significant amount of time on free software.

**APPRENTICE**

An apprentice is someone who has contributed in some way to a free software project, but is still striving to acquire the skills and standing in the community to make more significant contributions. Ideally, the Apprentice is in touch with either an individual mentor or a community that helps to gain these skills. An Apprentice spends a significant amount of time learning the craft of software development, whether by hands-on practice, academic study, or careful observation.

**SECTION C. ADVOGATO TRUST METRIC**  
(Source: <http://www.advogato.org/trust-metric.html>)

**ADVOGATO'S TRUST METRIC**

The basic trust metric evaluates a set of peer certificates, resulting in a set of accounts accepted. These certificates are represented as a graph, with each account as a node, and each certificate as a directed edge.

Advogato performs certification to three different levels: Apprentice, Journeyer, and Master. This is actually done by running the basic trust metric three times, using the "level" value in the certificate as a threshold. Thus, certification of Apprentices is computed using all certificates, while Master is computed using Master certificates only.

The computation of the trust metric is performed relative to a "seed" of trusted accounts. The core of the trust metric is a network flow operation. Informally, if there is a rich web of interconnections, flow reaches almost all the nodes.

**MAPPING INTO GRAPH**

The mapping of certificates into a graph is dependent on a parameter: the certification level  $l$ . Each account on Advogato corresponds to a node in the graph. An edge exists from node  $s$  to node  $t$  when account  $s$  has certified account  $t$  at level  $l$  or higher. In addition, there is a distinguished "seed" node, with predefined edges to accounts.

**ASSIGNMENT OF CAPACITIES**

The next step is to assign a capacity to each node in the graph. This is done by breath-first searching the graph from the seed, computing a shortest distance from the seed to each node. Then, capacities are assigned simply as a function of this distance. Nodes closer to the root have high capacity, which diminishes with distance. An example of the capacity assignment is shown below in Figure S1, although with much smaller capacities than used on Advogato:

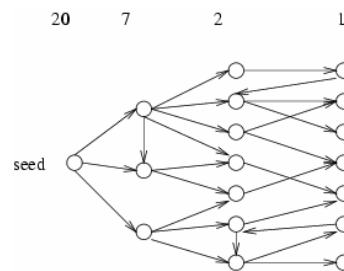


Figure S1. Advogato Trust Metric

**CONCLUSION**

The trust metric used in Advogato is easy to compute efficiently and fairly simple to understand. As such, it should find applications in security infrastructures, as well as defining online communities.