F-020

Intelligent and cooperative blog communities using MAS technology

Robin Swezey†, Masato Nakamura†, Shun Shiramatsu†, Tadachika Ozono†, Toramatsu Shintani†

1 Introduction

Our goal in this research is to provide with an architecture of a multi-agent system (MAS) capable of increasing the popularity of community blogs which are related in terms of affiliation between one another. The following example details our objective.

A pattern which is very likely to appear when Internet user λ starts a blog, is that of λ trying to find affiliates such as ν, sharing the same interests and ready to share his/her audience with λ. By engaging themselves in affiliation, λ and ν are likely to increase their number of visitors and page rank.

We propose a MAS architecture capable of providing automatization of this activity, in order to avoid them from dealing with time-consumings research and negotiation processes in affiliation. The Blog Pool, as an agent, will look for affiliate blogs and deal with the issues of equity in partnership, and propose the user to link them in their affiliates section.

1.1 Past research about blog agents

Some researchers have pointed out that blogs could be considered, in some way, as agents [1]. Blogs represent the individuals who initially set them up on the Internet, in order to share his knowledge and let his opinion be known around the world. Yet, the system lacks proactiveness and social behavior [2] to be defined as an intelligent agent.

1.2 Means of use

We define a blog b and its owner as the same entity.

b is expected to register on the system's homepage, with input data such as: the URI to b's main page, b's name, and a category selection. The blog will be affected a Blog Pool. To collect statistics, b must also include Javascript code on its template.

To look for new affiliates, b inputs the number of the position that he intends to attribute to his new affiliate in the Blogroll. For instance, if there are n places available in blog b's Blogroll and b wants an affiliate for the third slot from the top, b inputs 3.

† 名古屋工業大学大学院工学研究科　情報工学専攻

1.3 Affiliation utility function

V_b is the daily amount of unique visitors of b.

G_b is the Google page rank of b's main page.

We call by raw utility for b towards b' the function:

$RawU_b(b') = \frac{V_{b'}G_{b'}}{V_bG_b}$

This utility function is an indicator of the reative ranking, by visitors, for blog b in function of b'.

$U_b(b') = K \times RawU_b(b')$

$R_{b'}$ represents the amount of slots in b''s blogroll.

$r_{b'}(b)$ is one of the available slots for b in b''s blogroll.

$card(R_{b'}) \leq 5 \Rightarrow K = \exp^{-3\frac{r_{b'}(b)}{5-1}}$

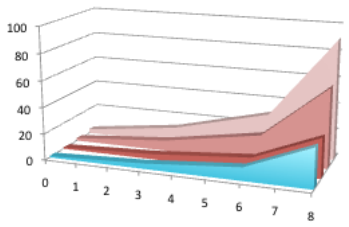
$5 < card(R_{b'}) \leq 10 \Rightarrow K = \exp^{-3\frac{r_{b_{b'}}(b)}{R_{b'})-1}}$

The topmost link in the Blogroll usually gets attention first, whereas the bottommost one gets attention last. We have chosen K as a decreasing exponential model because it is one of the patterns that can be observed in practice by monitoring.

Once an affiliate's $r_{b'}$ reaches five, the probability to get a click gets significantly lower ($\exp^{-3}$).

1.4 Proposed behavior model

We define expectation as the increase of new visitors that blogs wish for at every iteration (day). The behavior for blogs is such as expectation varies exponentially in function of their page rank. This can be explained easily when the evolution of page rank is roughly approximated to a logarithm of the number of visitors.



Patterns of visitors expected per iteration in function of page rank

The way our agents can handle expectation from blogs is by giving an average amount of new visitors per iteration for each page rank value as a basis. It

then applies an expectation coefficient and calculates the blog's expectation (blue curve: coefficient 1). The expectation will be matched against affiliation proposals with their utility values.

2 Society of Blog Agents

2.1 Pool layer

The Blog Pool Agents are agents that will allow us to actually evaluate the performance of our system. They hold a certain number of blogs, every one of them having a defined profile, and bearing expectations. They are profiled mainly by two parameters: expectations and activity rate. The latter is related to the tendency of bloggers to look for new affiliates. Each blog's initial number of visitors, page rank, expectation and other parameters are defined using normal distribution, around means and deviations that depend on the patterns we want to test.

In our simulation we affect Blog Pool Agents not by category but by domain (blogger, livejournal...). Their implementation and ways of extracting data from blog services will vary, but their interface to the MAS will remain the same.

Each blog in a Blog Pool Agent outputs a different satisfaction rate depending on its expectation for new visitors per day, and the actual number of new visitors per day it gets. We can then evaluate the system by measuring the average rate of satisfaction.

2.2 Negotiation layer

The Broker Agents are used by the Blog Pool Agents to find new affiliation opportunities. Each Broker Agent hold numerous blogs and their related data in their internal database, related to only one category. Depending on database needs for one category, other Broker Agents may be spawned to ensure load balancing.

When an Broker Agent finds enough potential affiliates for blog b in its internal database, it begins evaluating the utility function of b towards every element of the set of potential affiliates, and makes a first selection to limit the number of elements. When the system first start running, that selection is made randomly, but with time Broker Agents add scores to their utilities for blogs. The utility function is then evaluated from every element of the set towards b.

If the Broker Agent(s) responsible for b and b' reach the agreement that affiliation is possible given the place order proposed by b' for the link tag of b, it (they) will return a response the Blog Pool Agents of blogs b and b' with a partnership proposal to their bloggers in their respective affiliation homepage.

By using utility scores, propoals that get constantly rejected by b because of the same reason will not be shown again after some time.

Finally, the Test Coordinator Agent acts as a coordinator and collects data from Blog Pool Agents. We use it to monitor the runs.

3 Implementation

To implement this MAS architecture, we use the MiLog platform [3]. It provides with facilities to run agents and build flexible MAS systems using logic programming similar to Prolog, and numerous facilities such as message passing by CGI, mobile agent replication and transer, etc. MiLog runs in the Java environment and is easily scalable.

Since the system is meant to be used by a great amount of people, we intend to test it using sample data and patterns defined with normal distributions.

4 Conclusion

We have proposed a MAS architecture capable of assisting bloggers in finding adequate and equitable affiliations. Our system makes use of patterns and utility functions that closely resemble bloggers.

If the present research gets good results, it can be further extended to be applied to references (links that appear in "related links" sections) between articles of different blogs.

参考文献

[1] IPSJ Magazine Vol.48 No.3 Mar. 2007.

[2] Wooldridge, Jennings, "Intelligent Agents: Theory and Practice", 1995.

[3] N. Fukuta, T. Ito, T. Shintani, "MiLog: A Mobile Agent Framework for Implementing Intelligent Information Agents with Logic Programming", Proc. of the First Pacic Rim International Workshop on Intelligent Information Agents