Relation Grid: A Social Relationship Network Model

Jiaxing Song, Weidong Liu, Shaoyu Chen
Department of Computer Science & Technology, Tsinghua University, Beijing, China, 100084
{jxsong, liuwd}@tsinghua.edu.cn, csy98@mails.tsinghua.edu.cn

Abstract. In this paper, a social network model based on real social relationships, referred as Relation Grid, is proposed. Relation grid is isomorphic to the actual social network; hence it can represent the actual social network well. It is discussed that the proposed model is pure distributed, scalable, self-adaptive and self-organized. The application perspective of relation grid is encouraging. It not only provides a new way to acquire social relationships, but also can be used as an underlying background to construct online communities and organizations.

1 Introduction

In recent years, researches on the social network model become more and more popular. These researches can be categorized into several areas. One hot topic is Social Network Analysis (SNA) [4], which intends to discover the structure, manner of behaviors and individual character of members among groups, organizations and communities through analyzing the relationships and flows between them. The purpose of SNA is to improve the sharing, interaction and cooperation among groups, organizations and communities [5, 6]. Another topic is to utilize the concept and short the average distance of social networks, applying social network to other none-social network systems in order to improve the performance, scalability or availability of them [7, 8]. In addition, researches on the structure, operation and presentation of social network attract broad attentions [1, 9, 10].

However, current researches lean to focus on improving current systems using SNA method or social network concept while ignoring the value of the actual social network. In practical world, the social entities (people) are tied with relationships to constitute a social relationship network. Social relationships play an important role in social activities: in fact every social activity is conducted through the utilization of social relationships. Suppose that there is a mechanism which enables people establishing and acquiring social relationships and helps people to accomplish social cooperation through the Internet, it would greatly improve the information and knowledge sharing, interaction and cooperation between social entities. As far as we are concerned, there is no such mechanism which supports the actual social relationship representation, establishment and acquisition as well as interaction and cooperation upon them.

The Grid [2, 3] is a new and popular concept. The purpose of the grid is to integrate widely distributed, heterogeneous, autonomic computation and storage
resources to provide transparent and non-trivial services. The character of the grid includes resource distribution and sharing, highly abstracted, self-similarity, dynamic and various, autonomy and multiple administrations. In this paper, we originally proposed a mechanism which supports the actual social relationship representation, establishment and acquisition as well as interaction and cooperation upon them. Because the similarity in purpose and character of our model with the grid. We make use of the grid concept to refer our social relationship network model as Relation Grid.

The Relation Grid is constituted with nodes and relations, in which nodes represents people in actual society tied with relationships to form a social network. Apparently, the Relation Grid is isomorphic to the actual social network. We argue that the model we proposed is pure distributed, scalable, self-adaptive and self-organized. The application perspective of relation grid is encouraging.

The rest of this paper is organized as follows. Section 2 presents the Relation Grid model and concepts. Section 3 introduces the operations on Relation Grid. Section 4 discusses the scalability and other characters of Relation Grid. Section 5 presents the applications on Relation Grid. Finally, section 6 gives the summary and the future work.

2 The Relation Grid

The structure of relation grid is shown in Fig.1.

![Fig. 1. The structure of relation grid](image)

The relation grid is structural viewed as a weighted, directed graph which consists of vertexes (entities) and edges (relations). Let \( G = (V, E, a) \) be the notation for a graph G consisting of a set of vertexes \( V(G) \) and a set of edges \( E(G) \). \( a \) is a mapping from the set of edges \( E(G) \) to \((0,1]\), that is \( \forall r \in E(G) \), we say \( a(r) \) is the weight of edge \( r \).

The relation grid is consisted of the following elements:
Node: node is a vertex in graph $G$, a node $v \in V(G)$. Every relation is a binary relation between two nodes. A node usually contains Node Metadata, which describe the attributes of the node.

Relation: relation is an edge in graph $G$, a relation $r \in E(G)$. A relation is build between two nodes to represent the connection of them. $a(r)$ is the weight of edge $r$, written as $a$, $0 < a \leq 1$. It represents the closeness of a relation between two nodes and is referred as the correlation factor of a relation.

There are two types of relations. A Symmetrical Relation means that the relation between two nodes is symmetrical. If the relation between node $A$ and node $B$ is a symmetrical relation, we say the relation from node $A$ to node $B$ is $r_{AB} = \langle A, B \rangle$, then the relation from node $B$ to node $A$ is $r_{BA} = r_{AB} = \langle A, B \rangle$. An Asymmetrical Relation means that the relation between two nodes is asymmetrical. If the relation from node $A$ to node $B$ is asymmetrical relation, written as $r_{AB} = \langle A, B \rangle$, then the relation from node $B$ to node $A$ is $r_{BA} \neq r_{AB}$. We say $r_{BA}$ is the inverse relation of $r_{AB}$, written as $(r_{BA})^{-1} = r_{AB}$.

Relation Path: a directed path $P$ connecting two nodes is called the Relation Path between two nodes, $P = v_0r_0v_1r_1\cdots v_{l-1}r_{l-1}v_l$. Relation Path can be an abstracted concept with actual nodes disregarded, such as $P = \langle r_0, r_1, \cdots, r_{l-1} \rangle$. We say $l$ is the length of the relation path.

As shown in Fig.1, the relation path from node $K$ to node $I$ is $Kr_2Dr_1Ar_1Cr_2I$ or $\langle r_2, r_1, \cdots, r_2 \rangle$. We say $l$ is the length of the relation path.

Path Correlation Factor (PCF): the path correlation factor is defined as follows. Let relation path $P = \langle r_0, r_1, \cdots, r_{l-1} \rangle$, $a(r) = a_i$, $i = 0, 1, \cdots, l-1$, the PCF of relation path $P$ is $A = \prod_{i=0}^{l-1} a_i$. In Fig.1 the PCF of the relation path from node $K$ to node $I$ is $A = a_2a_1a_1a_2$.

3 Operations on the Relation Grid

Three types of operations are defined here to support the manipulation and functionality of the relation grid. Operations that define and maintain the structure of relation grid, called Definition Operation. Queries about the nodes, metadata and relations of relation grid, called Query Operation. Operations that support the sharing, interaction and cooperation of relation grid nodes, called Interaction Operation.

To dress all kinds of operations on relation grid, we introduce a Relation Grid Manipulation Language (RGML). The syntax of RGML is very similar with Structured Query Language (SQL). Discussions on RGML are beyond the scope of this paper, we would provide a detailed interpretation in succeeding papers.
3.1 Definition Operations

Definition operations of relation grid are mainly used to define and maintain the elements and structure of relation grid. Commonly, the definition operations include insertion, removal and modification of nodes, node metadata and relations, etc.

Each relation grid node is identified with a unique id. All nodes belong to the set of nodes. The node metadata is consist of (metadata.id, metadata.value) key-value pairs. A node can contain any pairs of metadata. Each relation has two attributes: the relations.id is the identifier of a relation; the relations.factor is the correlation factor of a relation.

The following are examples of definition operation on relation grid.

Insert a Node

```
INSERT INTO nodes SET id = "nodeID"
```

Modify Node ID

```
UPDATE nodes SET id = "newNodeID"
WHERE id = "oldNodeID"
```

Delete Node MetaData

```
DELETE FROM metadata
WHERE nodes.id = "nodeID" AND metadata.id = "metaDataID"
```

Modify an Asymmetric Relation

```
UPDATE relations
SET id = "relationID", factor = relationFactor
FROM nodes.id = "nodeID" TO nodes.id = "otherNodeID"
```

3.2 Query Operations

We provide various query operations on relation grid. Besides the basic query on nodes, metadata and relations, we provide several advanced queries, such as PCF Query, Relation Path Query and Nesting Query, etc.

The following are examples of query operation on relation grid.

Basic Query:

Query for Nodes

```
SELECT id FROM nodes WHERE nodes.id = "nodeID"
```

Query for Node MetaData

```
SELECT value FROM metadata
WHERE nodes.id = "nodeID" AND metadata.id = "metaDataID"
```

Advanced Query:

PCF Query

```
SELECT id FROM nodes WHERE PCF > somefactor
```

Relation Path Query

```
SELECT id FROM nodes WHERE relationPath = {relation1, relation2, relation3, ...}
```

Nesting Query

```
SELECT metadata.value FROM
SELECT * FROM nodes.metadata WHERE relationPath = {relation1, relation2, relation3, ...}
WHERE metadata.id = "metaDataID"
```
3.3 Interaction Operations

The main purpose of interaction operation is information sharing, interaction and cooperation between relation grid nodes. Interaction operation is an extendable operation which can be extended to support vast functions. Currently, we only provide a message dispatch operation. Before we introduce the interaction operation, we want to explain some basic concept.

*Message*: a message is the information and data sharing among nodes of relation grid. A message can be text, image, audio, video or the composition of them; can be a file, part of a file or composition of files and file parts. In one word, the message concept here is a general, abstract concept.

*Recursive Operation*: interaction operation with the same property can continually spread from one node to another without explicitly specified. We call this kind of operation a recursive operation. For example, node \( A \) wants to send a message to any nodes which is connected to it with relation \(<r_1>\), called \( r_1 \)-nodes, and to nodes which is connected to the \( r_1 \)-nodes with relation \(<r_1>\), and so on. We can use a recursive operation to implement this kind of functionality. In RGML a recursive operation is specified with the keyword *recursive*.

*TTL (time to live)*: a TTL sign indicate the total hops of an operation. If an operation spread from one node to another node through a relation, we call it a *hop*. So the TTL sign specifies the length of a relation path which is combined with the operation.

The following are examples of interaction operation on relation grid.

*Send a Message to the Node with PCF Larger Than Some Value and Total Hops Less Than 6*

```sql
SEND message TO
SELECT * FROM nodes WHERE PCF > somefactor and TTL < 6
```

*Send a Message Recursively to Nodes with “relation1”*

```sql
SEND message TO
SELECT * FROM nodes WHERE relationPath = {relation1}
```

recursive

4 Analysis and Discussion

In this section, we will present the quantitative analysis result on the scalability of relation grid, which is critical to the application of relation grid. We will also discuss the pure distributed, self-adaptive and self-organized characters of the relation grid.

4.1 Scalability Analysis

In relation grid environment, if the query and interaction operation spreads in an unrestricted flooding way, the overhead would be enormous.

Suppose the relation grid has \( n \) nodes, each node averagely is tied to other nodes with \( k \) relations. Averagely \( m \) relations are involved in operation on each node. We define selectivity factor as \( s = k / m \), where \( 0 \leq s \leq 1 \). Suppose the average hops of the
operation is \( h \), overhead of each hop is \( c \), and then the total cost of the operation would be:

\[
C = c \times m^h = c \times (k \times s)^h
\]  

(1)

It is easy to see that the total operation overhead increases exponentially with hops of the operation, which will remarkably impede the scalability of the relation grid.

However, according to the character of relation grid, we can take three ways to restrict the operation overhead while not inhibit the usability and availability of relation grid. They are **TTL Restriction**, **PCF Restriction** and **Relation Path Restriction**.

4.1.1 TTL Restriction

We can control the total hops of an operation through appending a TTL sign to it. This can effectively prevent the operation to spread without limit. Suppose the maximum hops of the operation is \( h_{\text{max}} \), as discussed in *W-S Small-World Network Model* [1], the average distance between two nodes in a small-world network is \( L \sim \ln(n) / \ln(k) \). Let \( h_{\text{max}} = \lambda \cdot \ln(n) / \ln(k) \), choose a reasonable \( \lambda \) to make sure that the operation would spread to the desired nodes with a high probability. The total cost of the operation would be:

\[
C_{\text{TTL}} = c \times (k \times s)^{\lambda \cdot \ln(n) / \ln(k)}
\]  

(2)

\[
\ln C_{\text{TTL}} = \ln c + \frac{\lambda \cdot \ln(n)}{\ln(k)} \ln(k \times s) = \ln(c \times \frac{\lambda \cdot \ln(n)}{\ln(k)})
\]

\[
C_{\text{TTL}} = c \times n^\varepsilon, \varepsilon = \lambda \cdot \frac{\ln(k \times s)}{\ln k}
\]  

(3)

\( \varepsilon \) is a constant depends on \( k \) and \( s \) and \( \lambda \).

We can see the complexity of operation with proper TTL restriction is at most the polynomial complexity.

4.1.2 Path Correlation Factor (PCF) Restriction

The spread of an operation on relation grid can be confined to a reasonable range by limiting the **PCF** (defined in Section 2). According to the small-world phenomenon, one node can find another desired node through the closest relations while only needs several hops. The **PCF** restriction will make operation only spread through the closest relations, so it wouldn’t impede the practicability of the operation on relation grid.

Suppose the average correlation factor of each relation involved in the operation is \( a_{\text{avg}} \), the total cost of the operation would be:

\[
C_{\text{PCF}} = c \times m^{a_{\text{avg}}} = c \times (k \times s)^{a_{\text{avg}}}
\]  

(4)
Given a reasonable PCF, we can guarantee that $A / a_{avg} \leq \lambda \cdot \ln(n) / \ln(k)$. According to (3), we can get that:

$$C_{PCF} \leq C_{Tr} = c \times n^s$$

(5)

The complexity of operation with proper PCF restriction is also at most the polynomial complexity.

4.1.3 Relation Path Restriction

The operations on relation grid are commonly restricted to certain relation path. The relation path restriction can remarkably confine the spread of operations and decrease the communication, computing and storage overhead.

Suppose we restrict the operation to certain relation path, the length of the relation path is $l$. Then the total hops of the operation $h$ would be $l$ if it’s not a recursive operation. The selectivity factor $s$ will change with each node along the relation path. Suppose the average selectivity factor is $s_{avg}$, the total cost of the operation would be:

$$C = c \times m_{avg} = c \times (k \times s_{avg})^l$$

(6)

The restrictions above can remarkably reduce the communication, computing and storage overhead of operations on relation grid while not impeding the functionalities of operations. In practical application, these restrictions are usually applied to an operation simultaneously to reduce the total overhead.

4.2 Discussion of Other Characters

We just demonstrated that the relation grid is scalable. Besides scalability, we argue that the relation grid is pure distributed, self-adaptive and self-organized.

*Pure distributed:* nodes of relation grid only have knowledge about itself and don’t have any knowledge about the whole network. Besides, there is no need for a global mechanism to organize the nodes. Nodes of relation grid are geologically distributed as well as logically independent. So the relation grid is pure distributed.

*Self-adaptive:* single failure of nodes would not cause the relation grid to be unavailable. Firstly, if one node is unavailable, there are always other paths to spread the operation, also, the application of relation grid commonly do not require every node which accord with the condition to be involved. The operations would work well if they can find desired nodes or relations with a high probability. The organization of relation grid is a loose-coupled organization; it would not require strict structural constraints. Therefore, the relation grid is self-adaptive.

*Self-organized:* the formation of relation grid is in a *bottom to top* way. The growth of relation grid is not organized or managed in a global mechanism. Instead, it is wholly depends on the dynamic changes of nodes and relations. Nodes can establish relations by self-studying, so we say the relation grid is self-organized.
5 Model Applications

Relation grid has a broad application perspective. On one hand, it provides a new way to represent, establish and acquire social relationship, on the other hand; it can be used as the underlying background to construct online communities and organizations which provide better support for information and knowledge sharing, interaction and cooperation among members of them.

5.1 Relation Acquisition

Thanks to the self-organizing character, the relation grid nodes can acquire and cumulate knowledge of other nodes which ties with relations to them. They can continually construct their personal network to constitute a growing, dynamic changing social network. It is well known that Google provide a new way to acquire knowledge through the internet. Similarly, the relation network provides a new way to acquire social relationships over the internet.

In practical society, the social entities (people) are tied with relationships to constitute a social relationship network. Social relationship plays an important role in social activities: in fact every social activity is conducted through the utilization of social relationships. Relation grid provides various query functions to support the relationship acquisition, here are some practical examples:

- Find my high school classmate. He is now in studying in Beijing. His family name is Chen.
  (relation: high school classmate; metadata: male, in Beijing, Chen)
- Find out all my acquaintance in order to tell them my new cell phone number.
  (relation: any; PCF > some value so that only acquaintance can be notified)
- Look for a friend or friend of friend who skills in C++, for there is a job position.
  (relation: friends, recursive; metadata: C++ skills)

5.2 Relation Grid Based Community

Network Community is a hotspot in researches on distributed computing. Generally, these network communities are online virtual communities consisted of entities which have similar interest and behavior characters. Virtual communities are hindered by the difficulties in establishing authentication procedures for identity, trust, and reputation. In virtual environments, personal characteristics are often blurred, indiscernible, or even deliberately faked.

Because the relations in relation grid is actual social relationship, the nodes would interact both online and offline, which make the relation between nodes more authentic and with more semantic meanings. This is how relation grid is distinct from normal virtual communities.

Fig.2 shows an online community built upon a relation grid. Nodes A, B, C are administrative nodes. Other nodes are tied to these nodes with certain conditions.