Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 24 (2013) 268 – 273

Procedia Computer Science

17th Asia Pacific Symposium on Intelligent and Evolutionary Systems, IES2013

Data Management with Flexible and Extensible Data Schema in CLANS

Shuai Wang[a,*], Yuanyuan Man[a], Tianyu Zhang[b], T. J. Wong[b], Irwin King[a]

[a]*Department of Computer Science and Engineering*
[b]*School of Accountancy*
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

Abstract

Data Management plays an essential role in both research and industrial areas, especially for the fields need text processing, like business domain. Corporate Leaders Analytics and Network System (CLANS) is a system designed to identify and analyze social networks among corporations and business elites. It targets to tackle some of difficult problems such as natural language processing, network construction, relationship mining, and it requires high-quality management of data. For data management, we propose a novel approach by integrating the essential XML files and auxiliary databases, with a flexible and extensible data schema. This data schema is the kernel of our data management. It achieves plenty of superiorities, namely, separability, scalability, traceability, distinguishability, version control and maintainability. In this paper, we specifically illustrate the data schema as well as the management approach in CLANS.

Keywords: Data Management; Data Schema; Flexible; Extensible; XML; Corporate Network; Social Network

1. Introduction

Data Management plays an essential role in both research and industrial areas[1,2]. It becomes a concerned issue among organizations since improper management of data could exert bad influences in various aspects. For instance, a terrible mechanism for organizing and managing data would cause trouble for developers, as it requires a large amount of modifications for every related programming model when something wrong with data happens, making the targeted system difficult to maintain[3].

Appropriate data management is particularly meaningful for qualitative data and it is the steadfast foundation for further data analysis[4]. That is because qualitative data, which stand for text, often contain potential pattern and information that could be applied for knowledge discovery, presented in many fields such as news, reviews and academic publications, also including the business domain that our project concentrate on.

* Corresponding author. Tel.: +852-6938-9466 ; fax: +852-2603-5024.
E-mail address: wangs@cse.cuhk.edu.hk

doi:10.1016/j.procs.2013.10.050

We present a new approach to deal with the issue of data management in CLANS. We have designed and implemented the Corporate Leaders Analytics and Network System (CLANS), a system designed to identify and analyze social networks among corporations and business elites in China. Given the power of relationship, the analysis of business social networks in China is of great significance[5]. However, there are still a number of difficulties existing in data collection, text processing, relationship mining and other technical issues. Thus, we develop CLANS to tackle some of those problems, by utilizing data mining and social computing related theories and techniques[6,7]. In this paper, we focus on discussing how we conduct data management in CLANS.

In terms of data management, the Extensible Markup Language (XML)[8] and the relational database management system (RDBMS) are two most pervasive and widely-used approaches, though managed data is a concept that can be implemented in variant ways[3]. While heated debates between XML and RDBMS still last, we realize that either of these two approaches has its own advantages and disadvantages. Since neither one is perfect, a hybrid approach would be a rational choice. So the question turns to be how to make full use of them comprehensively, and we give our answer based on the requirement of CLANS.

In CLANS, we propose a new approach by integrating auxiliary databases and united XML files with defined schema to achieve separability, scalability, traceability, distinguishability, version control and maintainability. In brief, we target at achieving flexibility and extensibility for the data management in CLANS.

Flexibility means that a system can handle different cases, even the unexpected ones. The data schema could never be pre-designed perfectly with the continuous development of a system, so we need to come up with a potable way to handle possible accidents. A typical example is, we define a NAME element for the entity Person, indicating the Chinese name for a certain person. However, it is possible to find out that a person hold another frequently-used name, like English name or Nickname, while collecting new data. It is unrealistic to neglect those correct names but instead we need to deal with them consistently. In this case the data schema is indeed required to be compatible and tolerable. We will discuss the solution in Section 3.4.

Extensibility requires that a system can manipulate new data easily, and extend its capability with future growth. In CLANS, we need to frequently collect and update information from the business area, taking account of its high turnover. Meanwhile, the data we collect are from diverse sources. Moreover, in the future, CLANS might also expand the research scope to some politically related organizations for further relationship mining as CLANS is our long-time project. They all raise more stringent requirement for the extensibility of our data schema.

The organization of the paper is as follows. We firstly present the data flow in Section 2. Then the management approach as well as data schema would be specifically discussed in Section 3. Finally, we draw the conclusion in Section 4.

2. Overview of Data Flow

The Figure 1 demonstrates the data flow as well as the procedure from multiple data sources to target XML files, before the stage of System Functionality (Part 4). It could be divided into three parts, Data Acquisition (Part 1), Data Preprocessing (Part 2), and Data Management (Part 3). These three parts are the data foundation in CLANS. Based on them we could conduct further advanced processing in the System Functionality, like data analysis, data mining and service providing, but these functionalities of CLANS are beyond the focus of discussion in this paper so we will not specifically elaborate them here.

Therefore, we only concentrate on the data management related components in this paper. Apart from System Functionality, Data Acquisition and Data Preprocessing are introduced in Section 2.1 and 2.2, and the Data Management is illustrated in Section 3.

2.1. Data Acquisition

We acquire data from various sources. We currently focus on the 2,500 Chinese listed firms and their senior managers. The information is mainly from China Securities Market and Accounting Research Database (CSMAR) and the web. On one hand, we collect data from the CSMAR, which contains all Chinese listed corporations, their senior executives and board members information from year 1999 to 2012; on the other hand, we crawl information

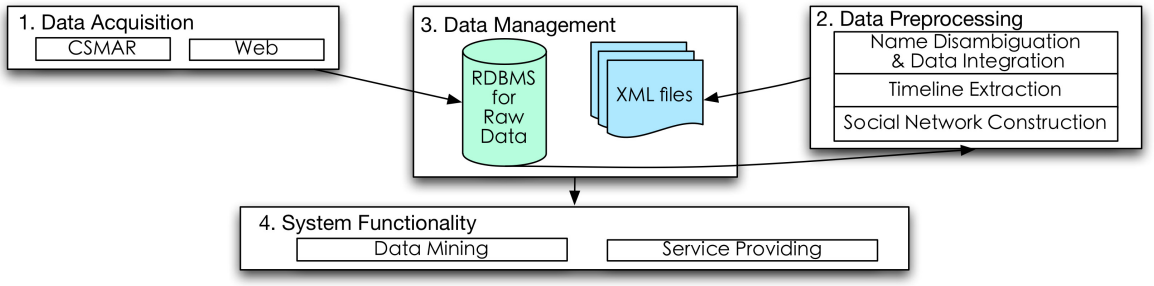

Fig. 1. Overview of the data flow

from websites like Baidu Baike[1] and Hexun Renwu[2]. Table 1 shows statistics of the current raw dataset. We keep on expanding and enriching our dataset.

Table 1. Statistics of the Raw Dataset.

Dataset	**Records**
CSMAR	350,000
Baidu Baike	43,926
Hexun Renwu	6,730

2.2. Data Preprocessing

For data preprocessing, we extract individual information from collected raw data and then construct the corporate social network. During this procedure, we confront problems like name disambiguation, data integration, natural language processing, social network detection and construction. To tackle these issues, we propose our solutions by applying records similarity matching, rule-based and HMM model[9], and formulate relations among individuals and corporations. However, we would not specifically illustrate the technical details here since this paper concentrates on the data schema and management.

3. Data Management

We propose a data management approach by integrating auxiliary databases and united XML files with the defined schema to achieve separability, scalability, traceability, distinguishability, version control and maintainability. A relational database management system and XML files are the major components of Data Management, as presented in Figure 1. On one hand, we apply databases to store different data from various sources and maintain them as raw data. That is, the raw data would never be taken placed by processed data, avoiding unrecoverable faults while merging them. On the other hand, we store the processed, united and latest version of data in XML files, providing the most updated information for the system.

3.1. Basic Idea

To achieve flexibility and extensibility, the substantial mission is to form a united and latest XML file from diverse and expanding databases for every individual. As shown in Figure 2, for a certain person P_i, all information about him is stored in different databases. For the first time creating the XML file for person P_i, we extract all the information of P_i from multiple sources and conduct data preprocessing. Eventually, we obtain a united data version for P_i

[1] http://baike.baidu.com/
[2] http://renwu.hexun.com/

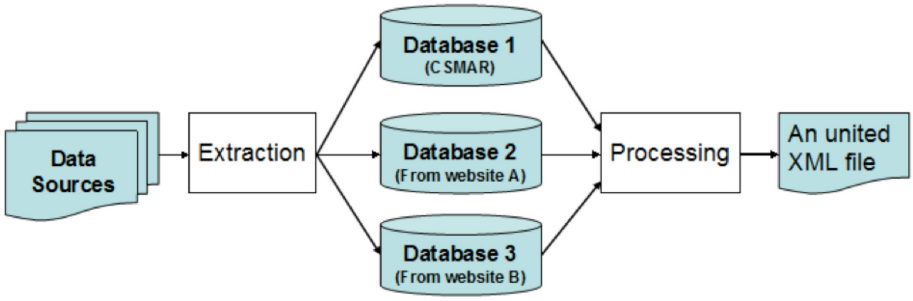

Fig. 2. Create one united XML file for one object.

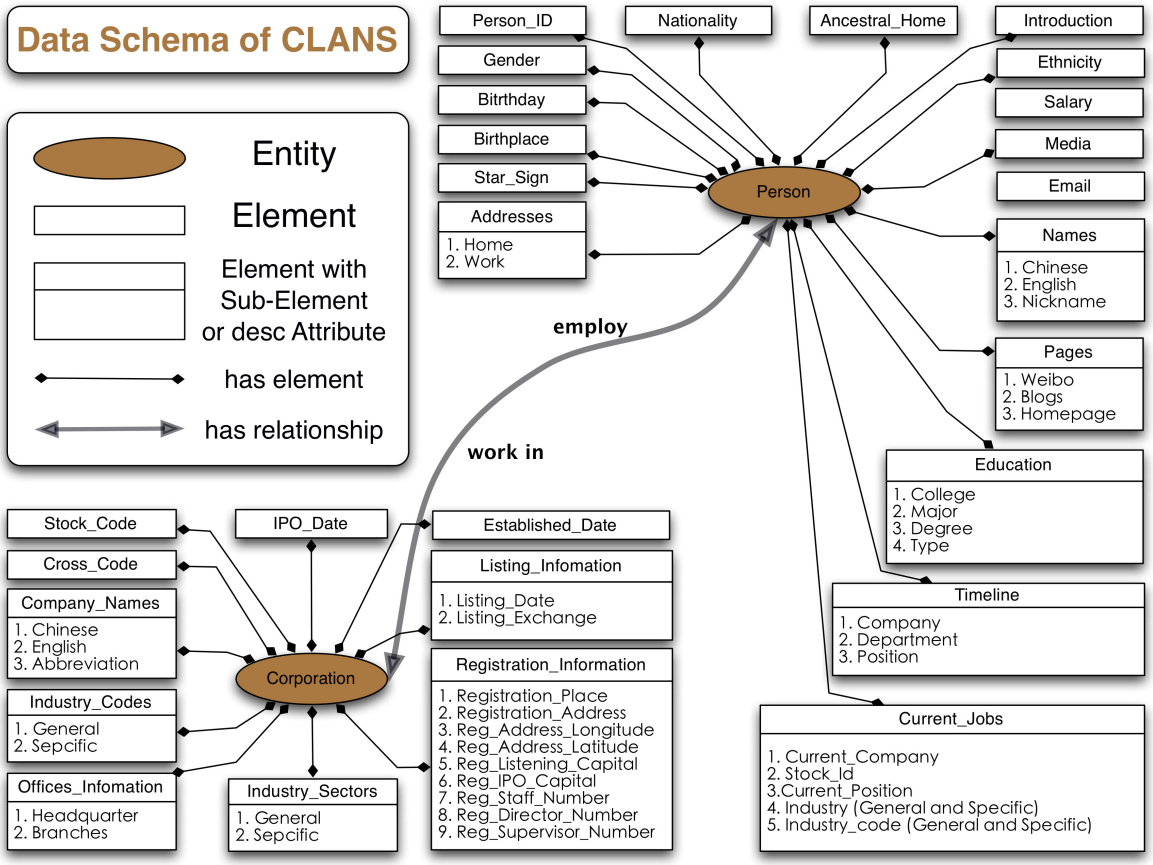

Fig. 3. The person and corporation entity schema.

in XML format. After that, every time we update the profiles, we just need to update the XML files with proper modification for the new collected data to achieve a new version, so that the XML files could always contain the latest and comprehensive information without inconsistency.

3.2. Core XML Files and Auxiliary Databases

We use XML files to store all united information for the individuals, and auxiliary databases to store different raw data from various sources. Table 2 exhibits statistics of the XML files. The XML files play essential roles in our data management, and the illustration would be followed with Section 3.3 and Section 3.4. By comparison, the relational database seems more assistant and easier for understanding, so we would not go further in the database issues. However, it is not the fact that the component of databases is ignorable or replaceable. The large data handling capacity, high efficiency, and powerful indexing functionality could not be easily taken place by XML.

Table 2. Statistics of the XML files.

Entity	Number of Instances
Person	83,929
Corporation	2,551

3.3. Entity Schema and Meaningful XML Attributes

In CLANS, person and company are two most vital entities. The defined schema is shown in Figure 3. For conciseness, some sub-elements and the *desp* (description, introduced in Section 3.4) attribute are not displayed.

In a XML file, the person or corporation is the root element, and its child elements are shown in Figure 3. The following XML sample displays a segment of a XML file. Besides the defined elements in the schema, we also propose meaningful attributes. The superiorities of our approach are explained with this sample in Section 3.4.

```
<person pid = ”27435”>
    <names>
      <name desc=”Chinese” src=”CSMAR_info” update=”128900000”>Tongming Wang</name>
      <name desc=”English” src=”Baidu_info” update=”134565800”>Tom Wang</name>
    </names>
    <gender src=”CSMAR_info” update=”128900000”>Male</gender>
    <birthday src=”CSMAR_info” update=”128900000”>1981-06-18</birthday>
</person>
```

3.4. Superiorities

Our approach achieves the following superiorities and makes data management flexible and extensible.

Separability. It is an implementation of MVC (Model-View-Control) model. The key concept is to separate the unified and the latest version of data representing format (the XML file) from the diverse and expanding databases. In this way, the back-end handler can keep crawling new data, and data management controller just need to concern about producing latest XML files, and subsequently the front-end can access latest united data with a formative way and conduct presenting, all of them decoupling.

Scalability. It is a lightweight operation to extend the XML content. For routine maintenance, we keep updating databases, especially in the business area with high turnover. Every time after we process and integrate the new acquired data, we do not need to reconstruct or create a new database. Instead, what we need is just to alter the existing XML files, adding some new features or just modifying selected fields that need updating. Compared to relational database, it is a lightweight operation. Most vitally, it is very easy to extend the XML content. If we extract a new feature from Web, e.g., <birthplace>, we just need to add one line in an XML file, e.g, <birthplace src=”HeXun_info” update=”164700000”>Hong Kong</birthplace>.

Traceability. With defined meaningful attributes, we make the modification of XML files traceable. For example, the *src* attribute indicates where the text value comes. For the example, the birthday element is from the table CSMAR_info while the English name is from Baidu_info. The *update* attribute records the timestamp we update an element. Both of them play important roles in the following version control.

Distinguishability. Utilizing defined attribute, it is easy to distinguish different elements, which belong to the same tag but have different meanings. For example, an individual might own *Chinese Name* and *English Name*. Thus,

we define the *desc* (description) attribute to distinguish different types of elements with a same tag. This attribute contributes the flexibility. If we extract a new name that was not pre-defined in the former schema, like *Nickname*, we just need to add a new line <name desc="Nickname">Tommy</name>.

Version Control. Combined with version control, our approach achieves error positioning, difference checking, and data recovering. With the help of meaningful attributes, we can easily check the data source and latest update time for every element in XML files. Thus, if an error were found, what leads to it (by *src* attribute) and when it happens (by *update* attribute) will be directly discovered. Further, we can find out what the specific false modification is, since the version control provides difference checking between two versions. Moreover, after the error detection, we can handle the emergency instantaneous, before we fix the system. What we need to do is just checking out the updated time (through *update* attribute) for that error, and then turn back to its previous version. Unlike database, on one hand, the proposed approach is to keep modifying as well as updating the existing XML files, not creating; on the other hand, the files are stored only by text, so our data management can be easily combined with version control mechanism. For our implementation, we apply Subversion (SVN) [3] to establish our own SVN server.

Maintainability. With all mentioned superiorities, the whole system becomes more maintainable. For the data acquisition, the developer can design a particular and suitable database for a targeted website accordingly, not constrained by the XML data schema due to the feature of separability. For the front-end developer, he might need to deal with new features appeared in new updated XML files with the schema evolving. However, since the schema is scalable, it is easier to handle a formative new feature than to execute complicated queries to different tables among databases. Moreover, with the help of version control, administrator as well as the controller feels much released to handle the system. If unexpected errors were detected from the new version of updated XML files, he could directly handle it without the help of the model developer and schema designer because he can apply former version (by version control) to maintain the stability of the system. Besides these typical examples, all the features of our data management guarantee the maintainability of the system.

4. Conclusion

Data Management plays an essential role in both research and industrial areas. It is especially vital in the fields that require qualitative data as well as text processing, including the business domain. CLANS is a system designed to identify and analyze social networks among corporations and business elites. So we propose a novel approach by integrating the essential XML files and auxiliary databases, with a flexible and extensible data schema. The data schema is the kernel of our data management, which achieves plenty of superiority, namely, separability, scalability, traceability, distinguishability, version control and maintainability, making a great contribution to CLANS.

Acknowledgements

This work was supported in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 413212).

References

1. Khatri, V., Brown, C.V.. Designing data governance. *Communications of the ACM* 2010;**53**(1):148–152.
2. Sockut, G.H., Iyer, B.R.. Online reorganization of databases. *ACM Computing Surveys (CSUR)* 2009;**41**(3):14.
3. Loh, A., van der Storm, T., Cook, W.R.. Managed data: modular strategies for data abstraction. In: *Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software*. ACM; 2012, p. 179–194.
4. Ryan, G.W., Bernard, H.R.. Data management and analysis methods 2000;.
5. Allen, F., Qian, J., Qian, M.. Law, finance, and economic growth in china. *Journal of financial economics* 2005;**77**(1):57–116.
6. King, I., Li, J., Chan, K.T.. A brief survey of computational approaches in social computing. In: *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE; 2009, p. 1625–1632.
7. Mo, M., King, I.. Exploit of online social networks with community-based graph semi-supervised learning. In: *Neural Information Processing. Theory and Algorithms*. Springer; 2010, p. 669–678.
8. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., Yergeau, F.. Extensible markup language (xml). *World Wide Web Journal* 1997; **2**(4):27–66.
9. Eddy, S.R.. Profile hidden markov models. *Bioinformatics* 1998;**14**(9):755–763.

[3] http://subversion.apache.org