Yuzuru Tanaka
Nicolas Spyratos
Tetsuya Yoshida
Carlo Meghini (Eds.)

Communications in Computer and Information Science 146

Information Search, Integration and Personalization

International Workshop, ISIP 2012
Sapporo, Japan, October 2012
Revised Selected Papers

Springer

Communications in Computer and Information Science 146

Yuzuru Tanaka Nicolas Spyratos
Tetsuya Yoshida Carlo Meghini (Eds.)

Information Search, Integration and Personalization

International Workshop, ISIP 2012
Sapporo, Japan, October 11-13, 2012
Revised Selected Papers

Springer

Volume Editors

Yuzuru Tanaka
Hokkaido University
Graduate School of Information Science and Technology
N-14 W-9, Sapporo 060-0814, Japan
E-mail: tanaka@meme.hokudai.ac.jp

Nicolas Spyratos
Université Paris-Sud
Laboratoire de Recherche en Informatique
Bât 650, 91405 Paris, Orsay Cedex, France
E-mail: nicolas.spyratos@lri.fr

Tetsuya Yoshida
Hokkaido University
Graduate School of Information Science and Technology
N-14 W-9, Sapporo 060-0814, Japan
E-mail: yoshida@meme.hokudai.ac.jp

Carlo Meghini
Consiglio Nazionale delle Ricerche
Istituto di Scienza e Tecnologie dell'Informazione
Via G. Moruzzi 1
56124 Pisa, Italy
E-mail: carlo.meghini@isti.cnr.it

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-40139-8 e-ISBN 978-3-642-40140-4
DOI 10.1007/978-3-642-40140-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013944644

CR Subject Classification (1998): H.3, H.5, H.4, H.2, I.2, K.4, J.1

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book contains the selected research papers presented at ISIP 2012, the 7th International Workshop on Information Search, Integration and Personalization. The workshop was held at Meme Media Laboratory, Hokkaido University, in Sapporo, Japan, during October 11–13, 2012. There were 29 presentations of scientific papers, of which 24 were submitted to the post workshop peer review. The international Program Committee selected 14 papers to be included in the proceedings.

The themes of the presented papers reflected today's diversity of research topics as well as the rapid development of interdisciplinary research. With increasingly sophisticated research in science and technology, there is a growing need for interdisciplinary and international availability, distribution, and exchange of the latest research results, in organic forms, including not only research papers and multimedia documents, but also various tools developed for measurement, analysis, inference, design, planning, simulation, and production as well as the related large data sets. There are similar growing needs for the interdisciplinary and international availability, distribution and exchange of ideas and works among artists, musicians, designers, architects, directors, and producers. These contents, including multimedia documents, application tools, and services are being accumulated on the Web, as well as in local and global databases, at a remarkable speed that we have never experienced with other kinds of publishing media. Large amounts of content are now already on the Web, waiting for their advanced personal and/or public reuse. We need new theories and technologies for the advanced information search, integration through interoperation, and personalization of Web content as well as database content. The ISIP 2012 workshop was organized to offer a forum for presenting original work and stimulating discussions and exchanges of ideas around these themes, focusing on the following topics:

- Information search in large data sets (databases, digital libraries, data warehouses)
- Comparison of different information search technologies, approaches, and algorithms
- Novel Approaches to information search
- Personalized information retrieval and personalized Web search
- Data analytics (data mining, data warehousing)
- Integration of Web services, knowledge bases, digital libraries
- Federation of smart objects

ISIP started as a series of Franco-Japanese workshops in 2003, and its first edition was placed under the auspices of the French embassy in Tokyo, which provided the financial support along with JSPS (Japanese Society for the Promotion of

Science). The workshops have alternated so far between Japan and France, and they have attracted increasing interest from both countries. Past ISIP workshops were:

2003 First ISIP in Sapporo (June 30–July 2, Meme Media Laboratory, Hokkaido University)
2005 Second ISIP in Lyon (May 9–11, University Claude Bernard Lyon 1)
2007 Third ISIP in Sapporo (June 27–30, Meme Media Laboratory, Hokkaido University)
2008 Forth ISIP in Paris (October 6–8, Tour Montparnasse, Paris)
2009 Fifth ISIP in Sapporo (July 6–8, Meme Media Laboratory, Hokkaido University)
2010 Sixth ISIP in Lyon (October 11–13, University Claude Bernard Lyon 1)

Originally, the workshops were intended for a Franco-Japanese audience, with the occasional invitation of researchers from other countries as keynote speakers. The proceedings of each workshop were published informally, as a technical report of the hosting institution. One exception was the 2005 workshop, selected papers of which were published in the *Journal of Intelligent Information Systems* in its special issue for ISIP 2005 (Vol. 31, Number 2, October 2008). The original goal of the ISIP workshop series was to create close synergies between a selected group of researchers from the two countries; and indeed, several collaborations, joint publications, joint student supervisions, and research projects have been initiated by participants of the workshop.

With these six workshops, the organizers concluded that the workshop series had reached a mature state with an increasing number of researchers participating every year. As a result, the organizers decided to open up the workshop to a larger audience by inviting speakers from ten countries to the ISIP 2012 Workshop.

The selected papers contained in this book are grouped into three major topics: knowledge federation and integration, social system support and visualization, and social information search and discovery. The first group of papers focuses on the theoretical and practical aspects of platform architectures or frameworks for the federation or integration of knowledge resources. The first two papers in the second group focus on social cyber-physical systems for optimizing or improving social system services based on big-data analysis and visualization. The other two focus on big-data analysis and visualization approaches for rice diseases and health care. The third group proposes theoretical foundations and methodologies for recommendation, statistical analysis, relevant knowledge extraction, and community discovery from social big data and text streams.

We would like to express our appreciation to all the speakers and participants of ISIP 2012 for their intensive discussions and exchanges of new ideas. This book is an outcome of those discussions and exchanged ideas.

March 2013

Yuzuru Tanaka
Nicolas Spyratos

Organization

ISIP 2012 was organized by the Meme Media Laboratory of the Hokkaido Unviersity, Sapporo, Japan.

Executive Committee

Co-chairs

Yuzuru Tanaka	Hokkaido University, Japan
Nicolas Spyratos	Paris-Sud University, France

Program Committee Co-chairs

Carlo Meghini	CNR-ISTI, Italy
Tetsuya Yoshida	Hokkaido University, Japan

Local Organization

Tetsuya Yoshida	Hokkaido University, Japan

Publication

Hajime Imura	Hokkaido University, Japan
Jonas Sjöbergh	Hokkaido University, Japan

Program Committee

Mina Akaishi
Hiroki Arimura
Vassilis Christophides
Jun Fujima
Kouichi Furukawa
Randy Goebel
Fumitaro Goto
Mohand-Saïd Hacid
Makoto Haraguchi
Hajime Imura
Kimihito Ito
Masahiko Itoh
Klaus Peter Jantke
Masaru Kitsuregawa
Dominique Laurent
Carlo Meghini
Shin-ichi Minato
Takafumi Noguchi
Yoshihiro Okada
Michel de Rougemont
Ken Satoh
Masahiko Sato
Michele Sebag
Nicolas Spyratos
Tsuyoshi Sugibuchi
Akihito Takano
Akio Takashima
Yuzuru Tanaka
Nigel Waters
Akihiro Yamamoto
Tetsuya Yoshida

Table of Contents

Knowledge Federation and Integration

Social System Support and Visualization

Social Information Search and Discovery

The Webble World in the Real World

A Possible Look at the Infant Stage of Future Web Development and Human Collaboration

Micke Kuwahara and Yuzuru Tanaka

Meme Media Laboratory, Hokkaido University, Sapporo, Japan
{mkuwahara,tanaka}@meme.hokudai.ac.jp

Abstract. During the last one and a half year that Webble World have been officially available to the public, the number of Webbles have more than doubled and the core platform have gone through numerous iterations of additional functions and improvements of existing ones. Now at the later part of 2012, Webbles are being used and developed for a multitude of exciting projects ranging from a Cancer research and trial manager, virtual lab and e-learning environments to an online sensor data visualization dash board and a nuclear research data editor. But Webble World it is not only about the big projects, it is equally as much, if not more, about the small home-made compound Webbles and tiny tools and leisure applications for personal improvement of the web experience that anyone can do on their own; and also there, Webble World have taken many steps forward with several mini games and web content displays. And of course, at the core of all this progress lies the new primitive Webble building blocks with their generic interfaces and inspiring features that have been developed by a constantly growing group of low layer developers from many parts of the world. If Webble World is here to stay, or if it is just another step towards the final realization of the knowledge federation meme media concept and the IntelligentPad technology, only time may tell, but the fact remains that it is happening now and we are along for the ride at the front seat.

Keywords: Meme Media Objects, Customize, Configure, Web, Share, Distribute, Resource, Interact, Participate, Federation.

1 Introduction

The concept of the Webble building block originates from the idea of the meme as coined by Richard Dawkins[4], which is the philosophical principle that all knowledge, culture and expressible thought can be seen as a gene of the mind which can evolve and survive through mutation, reproduction and adaptation just like ordinary genes, but instead of creating life forms it produces ideas. Also Mark Stefik's ideas on Knowledge Systems [5] have been highly influential when designing Webble World, as well as Huberman's ecology of computation [6].

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 1–10, 2013.

1.1 The Path to Modern Time

It all goes as far back as to the late 80's when the first paper on IntelligentPad [1] was released and the meme media laboratory was founded at Hokkaido University with one of many purposes to develop a software system that would allow users to objectify its digital environment and create blocks and pieces of standalone entities which would carry both content and behavior, and by being as small and as primitive as possible would enable a development of applications that would be far more adaptable and re-editable by the general public than anything known even today.

The result of that research and development resulted in the IntelligentPad[2][7][8][9], a framework environment that rested on the foundation that all single parts of an application was a functional application on its own, ranging from simple text input boxes and images to sliders, timers and advanced complex logic emitters, and that each part could be richly altered and edited as well as combined or recombined with other parts in order to form larger more advanced applications. Each part would be a pad-like box frame that could relate to other pads and be instructed on how to behave in relation to each other, henceforth the name IntelligentPad.

The architectural structure of such independent building blocks slowly took shape during the last decade and more, alongside with numerous Windows based versions of the framework, developed in cooperation with both strongly established partners as for example Fujitsu which built one version that was internally used for many years, but also together with successful venture companies as U.S. based K-Plex that built the still available PlexWare version from 2003 on the IntelligentPad concept.

1.2 The Path from IntelligentPad to the Webble

But in 2007 as the meme media laboratory joined an EU-based research project, ACGT, aiming to improve the infra structure of medical cancer research trials in Europe[10], with the idea to contribute to the project by building a PlexWare IntelligentPad based editor for medical trial management, the team realized that maybe PlexWare was not the best suitable option for developing such a tool. Reasons mentioned ranged from lack of multiplatform accessibilities and advanced rendering capabilities to stronger web integration and basic technological improvement. Finally it was decided that a new version of the IntelligentPad concept would be developed that could better match the needs of the ACGT project and future projects to come.

So between 2008 and 2010 extensive research on available technologies was conducted and multiple prototypes of a new web browser based software development tool was being produced until the team agreed on the specifics and details that finally would become the version presented within this paper. Even though most of the conceptual ideas and logical structures of IntelligentPad would still exist, numerous things would be improved and refined and due to that it felt natural to also evolve its name. So as this new Pad Enhanced Building Block Lifelike Entity took shape, as this Web based PEBBLE found its role, the name Webble became natural.

The goal was to develop a tool available directly inside any arbitrary browser or operative system which would federate and share humanity's collected knowledge during its constant progress and evolution by the reach of a mouse click. This framework is called Webble World [3][11].

2 Understanding the Webble

2.1 The Underlying Technology

The technology that in the end delivered most in terms of accessibility, development joy and stableness was Microsoft Silverlight. A vector graphic based web development framework primarily using C# and XAML and partly also HTML and JavaScript, which runtime component comes as a browser plug-in available for Both Apples OS X and Windows and also for most common browsers. Of course that choice of technology have not passed by completely uncriticized, especially by those who favors Linux, where latest Silverlight versions, four and five, completely lacks support, but also by those who favors Apples iPhone and iPad or Android based devices which does not allow any browser plug-ins. But in 2007/2008 when this decision was made, with Silverlight v1.0 soon to be v2.0 was running at full speed, the Android market was non existing and no iPads either, it was definitely a good and proper decision.

For those who falls within the target group Silverlight has been tremendous fun to use and to develop on and the result has been most impressive. Webble World have benefitted a lot by using this technology, not to mention the media rendering strength as well as strong support of multiple platforms which lessens the work of the web developer not having to be concerned about different browsers. So the current version of Webble World uses the full power of Silverlight and is available wherever Silverlight can be found (e.g. IE, Firefox, Safari, Opera, Chrome, Apple OS etc). What the future might hold regarding underlying technology will be discussed further on.

2.2 What Is a Webble?

A Webble is a Meme Media object, available inside a web-browser which have loaded the Webble World framework website.

The basic 'primitive' Webble have been developed by a web designer or programmer using traditional development tools, mainly Visual Studio and available templates, and then published (uploaded) to the Webble World repository via the website interface. What such a 'primitive' Webble can do is mainly restricted only by the skill of the developer.

After the Webble has been deployed it is then available by any website visitor who by using an internal Webble search engine can find appropriate Webbles for current need and download them into the Webble World work area found on the website.

'Primitive' Webbles can be edited and manipulated in many various ways, some common for all Webbles and some specific for certain Webbles. This editing may change the look and behavior of the Webble but also its content and relation to other Webbles. When 'Primitive' Webbles are instructed by the user to establish parent-child relationships with other Webbles and start sharing specific property values, different gadgets and widgets can be created directly in the browser without no or very little traditional scripting. Such a Webble group, known as a 'compound' Webble can then be saved online and made available by any other web user. Advanced and skilled Webble World visitors may use, manipulate and combine many such 'primitive' and 'compound' Webbles to form larger, more complex software packages which after developed and saved via the browser is considered a Webble 'application'.

'Primitive' Webbles evolves into 'compound' Webbles by each new user that interacts with it, and from a strain of a few 'primitives' may a large multitude of "mutated" 'compounds' be sprung, where those Webbles who are found and appreciated by most new users will survive and continue evolving.

Why Webbles?

As our understanding of culture- and knowledge sharing gains, along with the evolvement of today's Web to the interactive Web 2.0 and the semantic Web 3.0, the need for new, more open and more powerful tools emerges, tools that fill our need to build and communicate, to be inspired and further inspire others, to develop and change, edit and contribute, even without the formal education of a web developer. Such a tool needs to allow all types of users, from any background and with any sets of skills. It needs to be open and free so it may evolve alongside the rich source of knowledge it is carrying. It should be easy to access from anywhere in the world and it should be fully adaptable.

We believe Webble World could be such a tool.

2.3 What Constitute a Webble Structure?

The Webble is divided in two parts, a Model and a combined View/Controller part, plainly referred to as the View. These are two separate entities with similar structures but with different roles. The Model is considered to handle all internal matters that do not require any external interface, also known as the business logic, while the View deals mainly with all interaction with the user and holds all visual parts of a Webble. Any Model can be combined with any View at 'primitive' Webble design and development stage.

Next, in each of these parts one will find the concept of slots. A Slot is an externally available property parameter or method controller whose values may be viewed, exchanged, communicated and modified between present Webbles and also by users. The name slot tells us that we can see it as a hole or a plug where one may connect a contact in order to create a stream channel or path between two slots in two separate Webbles. This channel can be configured unilaterally or bilaterally.

In order to create a slot stream a Webble must first form a close relationship with another Webble. In Webble World that is known as a parent-child relationship where a structured hierarchy may be constructed with single Webble Parents having one or many children and those children may in turn be parent to one or many other Webbles, thus all together being a 'compound' Webble.

This is the baseline for every Webble.

Internal Design

Slot values can be of numerous types, from the classical numerals and strings, to more complex types as xml documents and object dictionaries. Slots can also be bounded to methods where the value of the slot can serve as method parameters. Slots can also be generated and bound to attributes of the visual objects in order to directly control the appearance of a Webble via the slot. Slots can be created, removed or reconfigured both at design time as well as at run time.

The internal slot values of a Webble can all be configured via platform provided configuration tools or forms and have their values changed by the user.

Slot communication is handled by the Webble via three ways of control methods (fig. 1). Whenever the parent have any value change in any slot it will fire the 'Update' message, informing every child that something have changed; it is then the task of the child to use the 'Gimme' message in order to retrieve the value of any specific slot from the parent to see if the update concerns them and the value of the slot they care for has changed. After that, it is then within the scope of the child to react upon the value collected. If instead it is the child slot that is altered it will transfer that alteration over to the parent with the 'Set' message to the connected parent slot, which in turn may make the parent react on the value change. Internally, both in the View and the Model, all slot changes fires a slot reaction method which sometimes does nothing and in other cases do a lot, maybe even start changing other slot values. A similar structure of communication is going on between the Model and the View within the Webble itself. This communication is client side only and does not work between Webbles on different computers in a shared network, but in theory such a communication could be implemented with the existing web service as a hub, but has not yet been so.

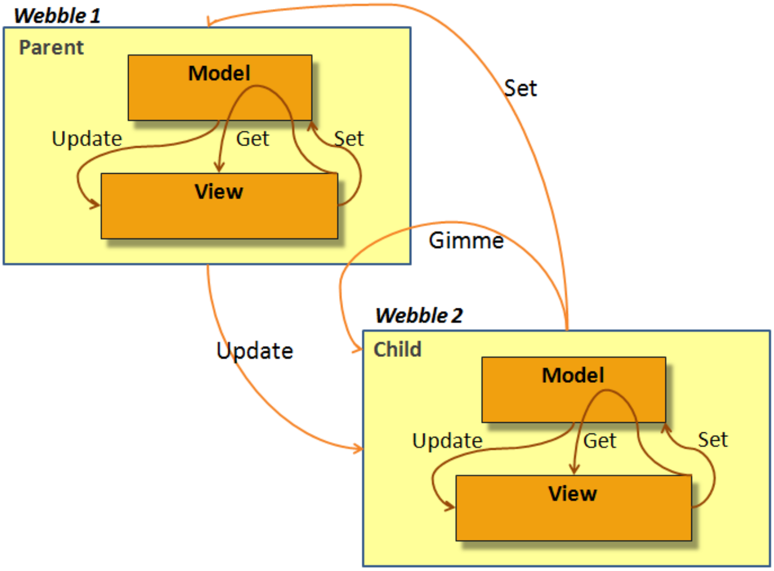


Fig. 1. A schematic of the internals of a Webble and its communication paths

In 'normal' slot communication a Webble can only have one slot channel open with each relative (parent or children) at one time. But there are ways to set up multiple channels by using specialized primitive Webbles like for example the Event Action

Manager Webble or the Slot Subscription Webble, which were made to handle that sort of structure.

A Webble can be duplicated freely, either as a separate entity or by sharing the Model with its original. In the latter case, a duplicate is called a shared copy. This is another form of internal communication between Webbles, which does not require any parent/child relationship and not only affects the slot values but the Model as a whole.

A Webble is not only defined as described above, with programming code. An even more important part of the Webble is the Webble definition or configuration file which is described in XML and hosts all internal values and properties of a particular Webble, like slot values and connections, children, model and much more depending on Webble class. It is this XML file that separates primitive Webbles (code generated only) from compound ones.

So in order to clarify; there are two levels of Webble construction, one is the creation of primitive Webbles which means to create some useful features in code and wrap them into a Webble, to then publish its executable online. The other level is the configuration of primitive Webbles, which under the hood means to create an XML definition file, but for the user means attaching and configuring Webbles together inside the browser environment with mainly basic mouse operations and then save the result online. In both cases the goal is to make the creation available for the public.

The User Operation Interface

No matter what Webble, there are some features and interfaces that basically all Webbles share. They all carry a context menu for basic Webble operations, like for example duplicate or slot manipulation. All will also display a border when selected which contains small interaction objects that allows the user to perform some common Webble operations like for example rotate or assign parent. When a feature is selected it is quite common that a form like window is opened that allow the user to edit or manipulate parts of the Webble or sometimes even the platform itself.

Most operations are done with mouse only, but in some cases the keyboard is needed for feeding values.

One of the major strengths of the Webble Meme Media Object is that though recognizable in structure and design as well as in human interaction interfaces, they are never limited to these only. A Webble can look, feel and behave any possible way, only limited by the imagination of the developer, but it will in the end always be what we expect it to be; just a Webble.

3 Progress

As for October 2012 there are now 45 primitive Webbles available from a total of 77 Webbles including compounds and applications too (115 if you also count private and hidden ones). These Webbles are mainly developed within the perimeter of Hokkaido University but there are developers and users from the whole world that contributes to the Webble World repository. The simplest primitive building blocks are text input boxes, images, timers, random number generators, sliders and alike and the more advanced and complex are system event handlers, interface manipulators, XML transformers, Bink-maps and such (fig. 2). By using these primitives numerous compound Webbles have been made that range from simple visually pleasing widgets and fun Webble experiments to fully working games and tools.

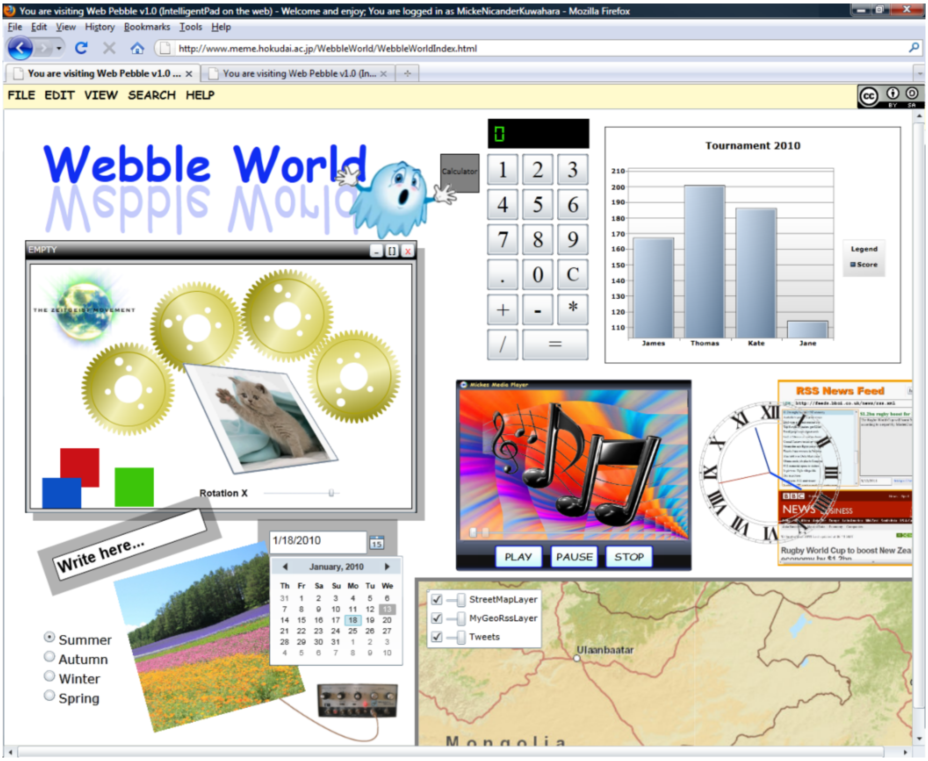

You are visiting Web Pebble v1.0 (IntelligentPad on the web) - Welcome and enjoy; You are logged in as MickeNicanderKuwahara - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.meme.hokudai.ac.jp/WebbleWorld/WebbleWorldIndex.html
You are visiting Web Pebble v1.0 ...
You are visiting Web Pebble v1.0 (In...
FILE EDIT VIEW SEARCH HELP
Webble World
EMPTY
THE ZEITGEIST MOVEMENT
Rotation X
Write here...
Summer
Autumn
Winter
Spring
Calculator
0
1 2 3
4 5 6
7 8 9
. 0 C
+ - *
/ =
Tournament 2010
210
200
190
180
170
160
150
140
130
120
110
James
Thomas
Kate
Jane
Legend
Score
Mickes Media Player
PLAY
PAUSE
STOP
RSS News Feed
BBC NEWS BUSINESS
Rugby World Cup to boost New Zea
1/18/2010
January, 2010
Th Fr Sa Su Mo Tu We
StreetMapLayer
MyGeoRssLayer
Tweets
Ulaanbaatar
Mongolia

Fig. 2. A wide range of Primitive and compound Webbles have been loaded into the Webble World work area running inside any common browser

Webbles are currently being used in at least 5 major ongoing projects and have been used successfully in 2 completed ones. One of these later application built in Germany under the name Solar Biker worked as a virtual lab environment for creating and experimenting with a solar energy driven model kit, and the other application developed under the umbrella of a EU project called Assets, worked as a tool for accessing information about museum artifacts and then structure relations between these artifacts and user generated contents and additional meta data. Those solutions which are still under development and undergoes continuous improvements are first one large tool for building, maintaining and analyzing cancer research trials, going by the name "Trial Outline Builder" which is a part of the extensive EU supervised project of ACGT and P-Medicine; another is the yet unnamed university course tool for visually teaching, for example, advanced algorithms, developed in Erfurt, Germany. A third is a tool developed in Japan for the international atomic energy agency (IAEA) which is aiming to investigate how atomic energy data may be analyzed and retrieved in a new useful way. The forth project is an exciting and so far successful attempt to port previous IntelligentPad solutions to the Webble World, a project that has brought many new inspiring primitive Webbles to the repository, opening up for a range of e-learning applications and entertaining games. Finally the current major project, going by the work name "Data analyzing Dashboard", aims to offer relevant users and data analysts a very versatile hands on tool for visualizing data correlations using a multiple range of Webble widgets ranging from tables, graphs and maps to data filter Webbles, xml manipulators and much, much more. Widgets which are easily transformed and remodeled or even additional developed and implemented due to the built in nature of Webbles. Such a tool is believed to highly increase the efficiency in decision making within many areas of society, but as initial focus is aimed towards snow removal in Sapporo, a city with annually 6m of snow, and disaster management in crisis situations (fig. 3).

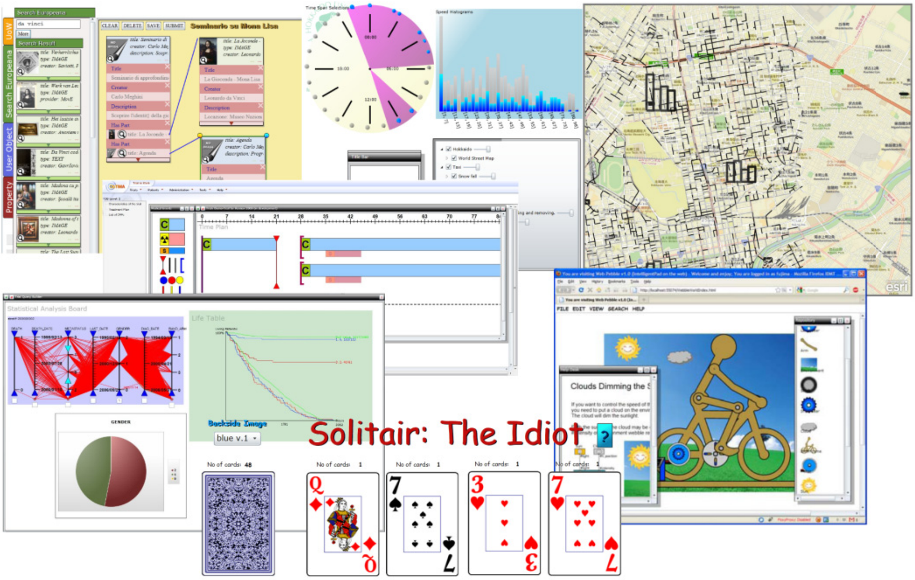


Fig. 3. A subset of existing Webble applications and projects currently being developed

The current rate at which new Webbles are introduced into the system is currently on average about 2-3 a month, but an indication, due to new projects, is that it might increase. But the general idea of Webbles is not that introduction of new primitives will increase indefinitely, but instead will reach towards some sort of peek and then stabilize into a slow stream of annual Webbles being added and old one being phased out when the building block repository contains enough pieces to satisfy most solutions and developers. But we are not there yet. If there are any limitations to what applications one can build in Webble World, we have not yet found them.

Being a Part of the Solution

To get to that point it is hoped for more independent users and Webble developers to enter the scene, and as more light gets on the Webble platform which seem to increasingly happen right now especially with the upcoming world's first Webble summit in Erfurt, Germany May 2013, that seems possible. This increased participation would also raise the flow of feedback and external input which whenever one works with human-driven interfaces is the engine for understanding and improvement.

The Future

Besides the projects mentioned earlier the future of Webble World lies a lot on understanding what users expect while working with Webbles and what features helps the flow of innovation, and which lack of features block it. So by compiling the list of feedback that is received it is the teams ambition to improve the current Webble World platform core to a state that invites more users and offers instant gratification for those who chose to use it. Areas like the Webble search interface will be improved but also internal core code that further simplify primitive Webble development will be implemented.

Another big, but important, area of concern is to make sure Webble World can be accessed on more devices. In reality that means implementing a Webble World version that does not require a browser plug-in, but instead runs mainly on HTML5 and JavaScript which at this stage is the technology with the largest range of support, and also the most promising commitment by other developers of both hardware and software. Such a Webble World is under design, but still have some major time left before it reaches a beta stage. The hope is to make any transition or parallel existence between current and future versions seamless.

We also constantly keep our eyes out for any other system similar to ours being released, but so far none of those pointed out has even been close to what Webble World can offer. They are either too limited in scope or too advanced and complex to use.

4 Summary

Webble World is a web top development and visualization tool based on meme media architecture that view all web content and its peripheral infra structure as standalone objects which can be manipulated, rearranged and personalized with the same ease one would edit a text document, with basic mouse operations and simple keyboard inputs. This system is available online since one year and is open and free for anyone who wants to try, individuals or organizations, at the following web address: http://www.meme.hokudai.ac.jp/WebbleWorldPortal/

The aim with the ongoing research and development of Webble World is to make software and application development more into a globally shared effort that involves people also outside the realm of programmers, as well as to minimize the time and workload of making software due to the capabilities to adapt, adjust, modify and combine freely already fully working and available building blocks.

The goal of creating a worldwide community of a Webble sharing population might maybe not just be around the corner but we believe its benefits would be so high that we are prepared to dedicate what it takes to get there.

References

1. Tanaka, Y., Imataki, T.: IntelligentPad: A Hypermedia System allowing Functional Composition of Active Media Objects through Direct Manipulations. In: Proceedings of the IFIP 11th World Computer Congress, San Francisco, USA, pp. 541–546 (1989)
2. Tanaka, Y.: Meme Media and Meme Market Architectures: Knowledge Media for Editing, Distributing, and Managing Intellectual Resources. IEEE Press & Wiley-Interscience (2003)
3. Kuwahara, M., Tanaka, Y.: Webble World, a web-based knowledge federation framework for programmable and customizable meme media objects. In: IET International Conf. on Frontier Computing, CP568, pp. 372–377 (August 2010)
4. Dawkins, R.: The Selfish Gene. Oxford University Press (1976)
5. Stefik, M.: Introduction to Knowledge System. Morgan Kaufmann (1995)
6. Huberman, B.A.: The Ecology of Computation. North Holland (1988)
7. Fujima, J.: A Unified Framework for Organizing, Accessing, and Federating Web Resources. Hokkaido University (2006)
8. Tanaka, Y.: Knowledge Federation over the Web, Based on Meme Media Technologies. In: Jantke, K.P., Lunzer, A., Spyratos, N., Tanaka, Y. (eds.) Federation over the Web. LNCS (LNAI), vol. 3847, pp. 159–182. Springer, Heidelberg (2006)
9. Tanaka, Y.: Meme media and a world-wide meme pool. In: Proceedings of the Fourth ACM International Conference on Multimedia, pp. 175–186. ACM (1996)
10. Weiler, G., Graf, N., Schera, F., Hoppe, A.: Ontology based data management systems for post-genomic clinical trials within a European Grid Infrastructure for Cancer Research. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007, pp. 6434–6437 (2007)
11. Kuwahara, M., Tanaka, Y.: Advanced “Webble” Application Development Directly in the Browser by Utilizing the Full Power of Meme Media Customization and Event Management Capabilities. In: IEEE International Conference on Multimedia and Expo, TEMPEKU Workshop: Tangible Edutainment Media for Playful Evolution of Knowledge and Understanding, ICME 2012, Melbourne, Australia, pp. 211–216 (July 2012)

Parallelism and Rewriting for Big Data Processing

Nicolas Spyratos and Tsuyoshi Sugibuchi

Laboratoire de Recherche en Informatique, Université Paris-Sud 11, France
Nicolas.Spyratos@lri.fr

Abstract. The so called "big data" is increasingly present in several modern applications, in which massive parallel processing is the main approach in order to achieve acceptable performance. However, as the size of data is ever increasing, even parallelism will meet its limits unless it is combined with other powerful processing techniques. In this paper we propose to combine parallelism with rewriting, that is *reusing* previous results stored in a cache in order to perform new (parallel) computations. To do this, we introduce an abstract framework based on the lattice of partitions of the data set. Our basic contributions are: (a) showing that our framework allows rewriting of parallel computations (b) deriving the basic principles of optimal cache management and (c) showing that, in case of structured data, our approach can leverage both structure and semantics in data to improve performance.

1 Introduction

Scientists regularly encounter limitations due to large data sets in many areas, including meteorology, genomics, complex physics simulations, and biological and environmental research. The limitations also affect Internet search, finance and business informatics. Data sets grow in size in part because they are increasingly being gathered by ubiquitous information-sensing mobile devices, aerial sensory technologies (remote sensing), software logs, cameras, microphones, radio-frequency identification readers, and wireless sensor networks.

The term "big data" refers to data sets with sizes beyond the ability of commonly-used software tools to capture, curate, manage, and process the data within a reasonable lapse of time [1]. As a consequence, what is considered big data varies depending on the capabilities of the organization managing the data set. For some organizations, facing hundreds of gigabytes of data for the first time may trigger a need to reconsider data management options. For others, it may take tens or hundreds of terabytes before data size becomes a significant issue.

Though big data is a moving target, as of 2008 limits were on the order of petabytes to exabytes of data [2]. However, size is not the only characteristic of big data. As stated in [3], big data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization.

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 11–20, 2013.

Moreover, the use of big data has drawn considerable criticism as well. Broader critiques have been leveled at Chris Anderson's assertion that big data will spell the end of theory, focusing in particular on the notion that big data will always need to be contextualized in their social, economic and political contexts [6].

Another criticism comes from the fact that, even as companies invest eight- and nine-figure sums to derive insight from information streaming in from suppliers and customers, less than half of the employees have sufficiently mature processes and skills to do so. To overcome this insight deficit, "big data", no matter how comprehensive or well analyzed, needs to be complemented by "big judgment", according to an article in the Harvard Business Review [7].

Moreover, consumer privacy advocates are concerned about the threat to privacy represented by increasing storage and integration of personally identifiable information; and expert panels have released various policy recommendations to conform practice to expectations of privacy [8].

Big data is difficult to work with using relational databases and desktop statistics and visualization packages, requiring instead massively parallel software running on tens, hundreds, or even thousands of servers [9][10][11].

Big data analytics, in particular, demands real or near-real time information delivery, and latency is therefore avoided whenever and wherever possible. With this difficulty, a new platform of big data tools has arisen, such as in the Apache Hadoop Big Data Platform [12] derived from papers on Google's MapReduce and Google File System (GFS). There is an impressive body of literature on MapReduce in the last few years but also some controversy coming mainly from the database community [14][15].

In this paper we present a new approach to parallel processing of big data called "RestrictReduce". Our approach proposes to combine parallelism with rewriting, that is *reusing* previous results stored in a cache in order to perform new (parallel) computations; and it is best suited when performing routine operations on big data that consist in:

– accessing each piece of data to retrieve a value
– then applying an aggregate function over all retrieved values

This process is shown schematically in Fig. 1(a), where D is the data set, V is a set of values, and f is a function that associates each element of D to a value in V. For example, D could be the set of identifiers of sales records and f the function that "extracts" from each record its dollar value. If we sum up all values of f, then we find the total sales; and we can call this process the "reduction" of function f with respect to the operation "*sum*". Similarly, if we take the average of all values, then we have the average sales; and this is another example of reduction of f with respect to "*average*". These values (total sale, average sales and so on) give summary characteristics of the data set, and provide valuable insights into the data.

However, if D is a big set, then parallelism is probably the only hope in order to achieve reasonable response times, and our RestrictReduce method is one way to achieve parallelism in two steps as follows:

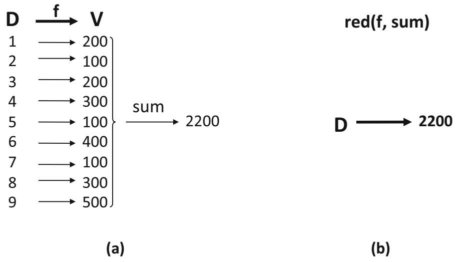


Fig. 1. A function $f : D \to V$ and its reduction with respect to the operation "sum"

restrict. restrict the function f to each block of a given partition of D
reduce. reduce each restriction of f to a single value (as explained above)

These two steps are illustrated in Fig. 2 and we shall explain them in more detail shortly.

Our approach is related to Map-Reduce in the following sense: one or more of the "smaller" reductions shown in Fig. 2 can be passed as a reduce task to MapReduce. However, MapReduce was designed by Google to handle use cases involving *unstructured* data, and in particular extracting words and phrases from webpages in order to create Google's web index. In other words, MapReduce was not originally designed to leverage structure in data; and as a consequence, its performance for processing structured data is not optimal. Unlike MapReduce, our RestrictReduce operator leverages the structure *and* semantics in data to obtain optimal performance (see section 5).

In the following section 2, we introduce some preliminary definitions and notations that we use to define our RestrictReduce operator. We then present our RestrictReduce rewriting method in section 3, and we outline an optimal cache management in section 4. In section 5, we discuss how optimal performance can be achieved in the case where the data set is a relational table with its semantics expressed in the form of functional dependencies; and in section 6 we offer some concluding remarks.

2 Preliminary Definitions and Notation

In order to define our approach we need some preliminary definitions and notation. We begin by defining what we mean by "reduction" of a function.

Definition 1 (Reduction)
Let $D = \{d_1, \ldots, d_n\}$ be a finite set and V any set. Consider a function $f : D \to V$ and an operation op over V. The reduction of f with respect to op, denoted red(f, op), is a pair defined as follows:

- $red(f, op) = (dom(f), op(\langle f(d_1), \ldots, f(d_n)\rangle))$

Figure 1 shows an example of reducing a function $f : D \to V$, where $D = \{1, \dots, 9\}$ and V is the set of integers. As "sum"is an operation over V, we can apply the above definition using "sum", to find that:

$red(f, sum) = (D, 2200)$

In Figure 1(b) we use the notation $D \to 2200$ instead of the pair $(D, 2200)$. In fact, we shall use the two notations, $(D, 2200)$ and $D \to 2200$, interchangeably to denote the result of reduction.

The operation *op* is called *reduction operation* and can be any operation applicable over V; for instance, if V is a set of integers (as in our example), the reduction operation can be "sum", "count", "max", "min", and so on. We shall often use the notation $red(f)$ instead of $red(f, op)$ whenever the reduction operation is understood from context.

Note that in the above definition of reduction we use the notation $op(\langle f(d_1), \dots, f(d_n)\rangle)$ to emphasize that all values of f must be taken into account, even if there are repeated values (i.e. even if $f(d_i) = f(d_j)$, for some $d_i \neq d_j$). For example, in Figure 1(a), although we have $f(1) = f(3) = 200$, the value 200 is taken twice into account when computing the sum of all values.

Definition 2 (Restriction over a partition)
Let $D = \{d_1, \dots, d_n\}$ be a finite set, V any set and $f : D \to V$ a function from D to V. Given a partition $\pi = \{D_1, \dots, D_n\}$ of D, the restriction of f onto π, denoted $res(f, \pi)$, is defined as follows:

- *$res(f, \pi) = \{f/D_1, \dots, f/D_n\}$, where f/D_i means restriction of f onto D_i*

Given a function $f : D \to V$ and a partition π of D, we can now define a new function, from π to V, by first restricting f onto π (i.e. to each block of π) and then reducing each restriction of f thus obtained.

Definition 3 (RestrictReduce)
Let $D = \{d_1, \dots, d_n\}$ be a finite set, V any set and $f : D \to V$ a function from D to V. Given a partition $\pi = \{D_1, \dots, D_n\}$ of D, the RestrictReduce operation on f and π, denoted $RR(f, \pi)$, is a function from π to V defined by:

- *$RR(f, \pi)(D_i) = red(res(f/D_i))$, $i = 1, \dots, k$*

The term "RestrictReduce" comes from the order in which the operations "restrict" and "reduce" are applied to each D_i.

Figure 2 shows the application of RestrictReduce to the function f of Figure 1, using the partition $\pi = \{D_1, D_2, D_3, D_4\}$ of D, where $D_1 = \{1, 2\}$, $D_2 = \{3, 4\}$, $D_3 = \{5, 6, 7\}$, $D_4 = \{8, 9\}$. The result of applying RestrictReduce to f and π is a function from $\{D_1, D_2, D_3, D_4\}$ to V, as shown in the figure.

It is important to note that, as seen in this example, the RestrictReduce operation takes as input a function f, and a partition π of the domain of definition of f, and returns a function $RR(f, \pi)$ with domain of definition π and with the same co-domain as f. In other words, it is important to note that:

- the input of RestrictReduce is the function $f : D \to V$
- the output of RestrictReduce is the function $RR(f, \pi) : \pi \to V$

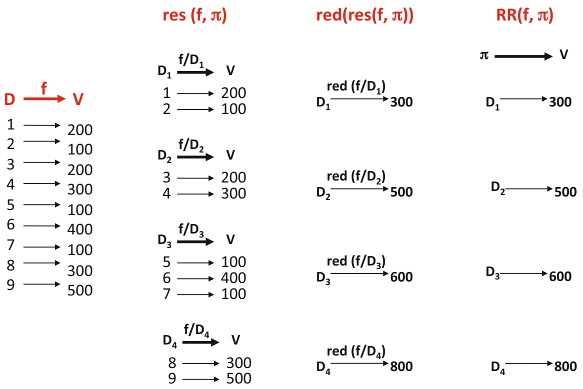


Fig. 2. Application of RestrictReduce to the function f of Fig. 1

Note that we can give an alternative (but equivalent) definition of the RestrictReduce operator, by introducing an extended form of reduction, applicable to a set of functions with common co-domain (assuming the same reduction operation for all functions involved). More precisely, given a set of k functions $g_i : S_i \to W, i = 1, \ldots, k$, let us use the notation $red(g_1, \ldots, g_k)$ to mean the set of reductions $\{red(g_1), \ldots, red(g_k)\}$, with respect to some operation (the same for all functions). Then the RestrictReduce operator can be seen as the composition of functions red and res: $RR(f, \pi) = red \circ res$.

Several important remarks on the RestrictReduce operator are in order here. First, since the output of RestrictReduce is a function, we can apply RestrictReduce to it again (if necessary).

Second, if we reduce the result of RestrictReduce (i.e. if we reduce the function $RR(f, \pi)$ then we obtain the reduction of f. It follows that we can replace the computation of the reduction of f by a number of reduction computations on “smaller” functions; and that we can eventually assign each of the smaller reductions as a “reduce task” to be performed by a different processor, thus “parallelizing” the computation. Clearly, each processor can then apply RestrictReduce recursively.

Third, note that there are two extreme partitions of D, namely the *coarse partition* $\{D\}$, consisting of a single block; and the *fine partition* $\{\{d\}/d \in D\}$, consisting of as many singleton blocks as there are elements of D. Using the coarse partition in a RestrictReduce computation is tantamount to performing a direct computation of reduction (i.e. without partitioning D); and using the fine partition is tantamount to performing no reduction at all (i.e. the result of reduction is the function f itself).

Fourth, it is well known that the set of all partitions of a set D can be ordered as follows: $\pi \leq \pi'$ if each block of π is included in a block of π'. Under this ordering we have $F \leq \pi \leq C$, for all partitions π, where F is the fine partition and C is the coarse partition. Note that $red(f) = red(f, C)$, and this justifies the notation $(D, 2200)$, or $D \to 2200$ that we introduced in Figure 1 for representing the reduction of f.

Fifth, although what we said so far is true for any partition π, it is usually the application at hand that actually “imposes” a partition. Indeed, suppose again that D is the set of identifiers of all sales records in a big company with four stores. It is very likely that each store manages its own sales records by storing them in a local database. In this case, D is *de facto* partitioned into four subsets, each subset residing in the local database of one store. It is therefore the actual partition defined by the stores that will be used in the computations. By the way, in this setting, computing the total sales of the company based on this partition, will require communication between the databases of the four stores, as well as a certain degree of synchronization. However, such considerations lie beyond the scope of the present paper.

As a final remark, if D resides in a single, central repository, then any partition of D can be used to compute total sales; and the question is which partition to choose in order to speed up computation. One factor influencing this choice is the number of available processors and their speeds. For example, suppose that there are only two processors available, one being twice as fast as the other. In this case, clearly, we should partition D into two blocks, one being twice as big as the other; and we should assign the bigger block to the faster processor and the smaller to the slower processor.

3 RestrictReduce Rewriting

In this section we present a basic result, namely how to rewrite one RestrictReduce computation in terms of another, whose result has been stored in a cache.

First, let us explain the basic idea using the example of Figure 2. Think again of D as being the set of identifiers of all sales records of a company having four stores, and let D_i be the set of sales records of store i, $i = 1, 2, 3, 4$. Then $\pi = \{D_1, D_2, D_3, D_4\}$ is a partition of D, and if we apply RestrictReduce on f based on π, we will find the result shown in Figure 2. As we have explained in the previous section, this result is a function, denoted $RR(f, \pi)$. As shown in Figure 2, this function associates each block D_i of π with a value in V and therefore $RR(f, \pi)$ represents the total sales by store.

Next, suppose that stores 1 and 2 are located in Lyon and stores 3 and 4 in Paris, and that we would like to find the total sales by city. Clearly, to do this we have to apply RestrictReduce on f, but this time based on the partition $\pi' = \{D_L, D_P\}$, where $D_L = D_1 \cup D_2$ is the set of sales records in Lyon and $D_P = D_3 \cup D_4$ is the set of sales records in Paris.

However, instead of passing again over all sales records of D we can simply apply RestrictReduce to the result of the previous RestrictReduce application, based on a partition π/π' of π defined as follows: $\pi/\pi' = \{\{D_1, D_2\}, \{D_3, D_4\}\}$. Indeed, it is easy to see that, by applying RestrictReduce on $RR(f, \pi)$, based on π/π', we find the desired result (i.e. total sales by city). Intuitively, what we do is sum up the totals by store for each city. Formally, this is done *not* by using f but by *reusing* $RR(f, \pi)$, which is the total sales by store.

The key observation here is that we were able to reuse $RR(f, \pi)$ because each block of π is included in a block of π'. Hence the following definition of quotient partition that leads to our basic theorem of RestrictReduce rewriting.

Definition 4 (Quotient Partition)
Let D be a finite set and let π, π' be two partitions of D. If $\pi \leq \pi'$ then the quotient of π by π', denoted π/π', is a partition of π defined as follows:

1. *for each block D' of π' the collection $\{D \in \pi | D \subseteq D'\}$ is a block of π/π'*
2. *there is no other block in π/π'*

To see that π/π' is indeed a partition of π it is sufficient to observe that (a) every block of π is in some block of π/π' and (b) the blocks of π/π' are pairwise disjoint. The quotient partition π/π' is the key concept of our RestrictReduce rewriting as it is stated in the following theorm.

Theorem 1 (RestrictReduce Rewriting)
Let D be a finite set and let π, π' be two partitions of D. If $\pi \leq \pi'$ then $RR(f, \pi') = RR(RR(f, \pi), \pi/\pi')$

Rewriting is a powerful tool for "boosting" performance of RestrictReduce.

To see the kind of savings in computation time, assume that D contains one million sales records made up from 250000 sales records per store. Then computing $RR(f, \pi')$ directly from f (i.e. *without* rewriting) will require accessing one million records, while computing $RR(f, \pi')$ *with* rewriting (i.e. using the already computed and stored function $RR(f, \pi)$ will require accessing just four records (i.e. those that make up the function $RR(f, \pi)$. Of course the rewriting of RestrictReduce computations incurs extra cost, namely (a) the cost of storing old results and (b) the cost of determining whether $\pi \leq \pi'$. A detailed account of the compromise between these two factors lies outside the scope of this paper. In the remaining of this paper, we simply discuss the basic principles underlying such a compromise.

The cost of storing old results depends on the application area and the specific cache management strategy used (e.g. continuous query caching). This topic lies outside the scope of the present paper. As for the cost of determining whether $\pi \leq \pi'$, this can be done using the semantics of the application. For instance, in our previous example of four stores located in two cities, it was obvious that $\pi \leq \pi'$, as each store is located in one and only one city. Therefore, the set of sales records of each store is contained in the set of sales records of the city in which the store is located. However, in general, determining whether $\pi \leq \pi'$ is not an easy task - and in fact, section 5 of this paper is devoted precisely to this question, in a particular setting. First though let us outline the basic principles underlying optimal cahe management in our approach.

4 Cache Management for RestrictReduce Rewriting

In view of our discussion in the previous section, if $RR(f, \pi)$ is already stored in a cache then any new RestrictReduce computation $RR(f, \pi')$ with $\pi \leq \pi'$

can be rewritten as $RR(RR(f, \pi), \pi/\pi')$. However, to begin with, we have to answer the following basic question: which RestrictReduce computations should be stored in the cache? It follows from the previous remark that the smaller the partition π in $RR(f, \pi)$ the more the possibilities of rewriting new RestrictReduce computations $RR(f, \pi')$ in terms of $RR(f, \pi)$. However, the smaller the partition π, the larger the cardinality of π and therefore the larger the size of $RR(f, \pi)$. In the end, it all boils down to a compromise between the size of the function $RR(f, \pi)$ to be stored and the available cache size. Therefore, assuming $RR(f, \pi)$ is already stored in the cache, here is how an incoming RestrictReduce computation $RR(f, \pi')$ should be handled:

```
if there is $RR(f, \pi)$ in the cache with $\pi \leq \pi'$ then
  rewrite $RR(f, \pi')$ as $RR(RR(f, \pi), \pi/\pi')$
  $result \leftarrow$ evaluate $RR(RR(f, \pi), \pi/\pi')$
else
  $result \leftarrow$ evaluate $RR(f, \pi')$
  $buffer \leftarrow$ accumulate all $RR(f, \pi)$ such that $\pi' \leq \pi$
  if $resul$t fits in cache after deleting $buffer$ then
    delete $buffer$ from cache and store $result$ in cache
  end if
end if
return $result$
```

Regarding the if-branch above, the question is what happens if the cache contains two results, say $RR(f, \pi_1)$ nd $RR(f, \pi_2)$ such that $\pi_1 \leq \pi'$ and $\pi_2 \leq \pi'$. In this case, clearly, we have every interest to choose the $RR(f, \pi_i)$ with the smaller cardinality of π_i. As for the replacement of "buffer" by "result" in the else-branch it is justified by the fact that each $RR(f, \pi)$ can be rewritten in terms of $RR(f, \pi')$ and therefore it no more needs to be stored. Finally, it is important to note that, following the above algorithm, the cache always stores results whose partitions are pairwise incomparable and minimal (i.e. cache management is optimal). A detailed account of cahe management in specific application areas is given in a forthcoming full paper.

5 Leveraging Structure and Semantics of the Data Set

As we mentioned in the introduction, MapReduce was not originally designed to leverage structure in data, and as a consequence its performance for processing structured data such as relational data is not optimal.

One of the main features of our RestrictReduce operator is that, unlike MapReduce, it can leverage structure *and* semantics in the data set, in order to improve performance. To see this, let us assume that the data set is a relational table $D(Tid, A_1, \ldots, An)$, where Tid stands for "tuple identifier" and $A_1, \ldots, A_n$ are the attributes of the table. Moreover, let X be a set of attributes and let pr_X denote the projection of D over X.

In this case, determining whether $\pi \leq \pi'$ can be done using the functional dependencies satisfied by D. Indeed, as noted in [17], each projection pr_X induces a partition π_X of the set of tuple identifiers of D as follows: if x is an X-value in pr_X then the set of all identifiers in D whose corresponding X-value equals x form a block B_x of π_X; therefore $\pi_X = \{B_x / x \in pr_X\}$. The following proposition states a fundamental relationship between partitions and functional dependencies over the table D.

Proposition 1 (Deciding Partition Ordering)

- *(a)* $\pi_X \leq \pi_Y$ *iff* $X \rightarrow Y$
- *(b) if* $X \supseteq Y$ *then* $\pi_X \leq \pi_Y$

As an example, consider the table $D(Tid, Date, Store, Item, ItemCategory, ItemSupplier)$ and assume that the following functional dependencies are satisfied among the attributes of D:

- $Store \rightarrow StoreCity$
- $Item \rightarrow ItemCategory, ItemSupplier$

As we explained earlier, the totals by store can be computed by a RestrictReduce computation using the partition π_{Store}. The result is the function $RR(f, \pi_{Store})$. Now, as the dependency $Store \rightarrow StoreCity$ holds, it follows from Proposition 1(a) that $\pi_{Store} \leq \pi_{StoreCity}$. Therefore, if we compute the function $RR(f, \pi_{Store})$, and store it in a cache, then (by Theorem 1) we can compute the totals by $StoreCity$, as follows:

- $RR(f, \pi_{StoreCity}) = RR(RR(f, \pi_{Store}), \pi_{Store}/\pi_{StoreCity})$

Similarly, if we compute (and store) the totals by $Item$ then (as $Item \rightarrow ItemCategory, ItemSupplier$ holds) we can compute the totals by $ItemCategory$ and by $ItemSupplier$ as follows:

- $RR(f, \pi_{ItemCategory}) = RR(RR(f, \pi_{Item}), \pi_{Item}/\pi_{ItemCategory})$
- $RR(f, \pi_{ItemSupplier}) = RR(RR(f, \pi_{Item}), \pi_{Item}/\pi_{ItemSupplier})$

6 Concluding Remarks

In this paper, we have proposed a novel approach to the parallel processing of big data, called RestrictReduce, as well as a rewriting technique that allows reusing previous RestrictReduce results. We have given a sufficient condition for the rewriting in our approach. Finally, we have seen how we can improve performance in the case where the data set is a (big) relational table with its semantics expressed in the form of functional dependencies.

We are currently experimenting with RestrictReduce in order to get more insight into the rewriting method and its performance with real data sets. Future work includes (a) a detailed study of complexity of our rewriting method, (b) the definition of a high level language for big data analytics and (c) the applicability of our approach to volatile data sets. Concerning this last point, we consider an update changing the data set D to a data set D', and we ask the following: given the result r of a RestrictReduce computation on D, how can we compute the result r' of the *same* RestrictReduce computation on D' by performing a RestrictReduce computation only on $D \setminus D'$ (i.e. only on the new data).

References

1. Manyika, J., Chui, M., Bughin, J., Brown, B., Dobbs, R., Roxburgh, C., Byers, A.H.: Big Data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute (May 2011)
2. Horowitz, M.: Visualizing Big Data: Bar Charts for Words. Wired Magazine 16(7) (June 2008)
3. Douglas, L.: The Importance of 'Big Data': A Definition. Gartner (June 2012)
4. Data, data everywhere. The Economist (February 25, 2010)
5. Executive Office of the President: Big Data Across the Federal Government. White House (March 2012)
6. Graham, M.: Big data and the end of theory? The Guardian (March 9, 2012)
7. Shvetank, S., Horne, A., Capellá, J.: Good Data Won't Guarantee Good Decisions. Harvard Business Review (September 2012)
8. Ohm, P.: Don't Build a Database of Ruin. Harvard Business Review (August 23, 2012), http://blogs.hbr.org/cs/2012/08/dont_build_a_database_of_ruin.html
9. Jacobs, A.: The Pathologies of Big Data. ACM Queue (July 6, 2009)
10. Monash: eBay's two enormous data warehouses. DBMS2 (April 30, 2009), http://www.dbms2.com/2009/04/30/
11. Monash, C.: eBay followup — Greenplum out, Teradata > 10 petabytes, Hadoop has some value, and more. DBMS2 (October 6, 2010), http://www.dbms2.com/2010/10/06/
12. Apache Hadoop project page, http://hadoop.apache.org/
13. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. In: 6th Symposium on Operating Systems Design and Implementation, OSDI 2004, Sponsored by USENIX, in Cooperation with ACM SIGOPS, pp. 137–150 (2004)
14. DeWitt, D.J., Stonebraker, M.: MapReduce: A major step backwards. Vertica The Database Column (January 17, 2008)
15. Bain, T.: Was Stonebraker right? (September 15, 2010), http://blog.tonybain.com/tony_bain/2010/09/was-stonebraker-right.html
16. Ferrera, P., de Prado, I., Palacios, E., Fernandez-Marquez, J.L., Di Marzo Serugendo, G.: Tuple Map Reduce: Beyond classic MapReduce. In: IEEE Intl. Conf. on Data Mining, ICDM 2012, Brussels (December 2012)
17. Spyratos, N.: The Partition Model: A deductive Database Model. ACM Transactions on Database Systems 12(1), 1–37 (1987)

Toward Cost-Aware Semantic Caching in the Cloud

Laurent d'Orazio[1], Dominique Laurent[2], and Nicolas Spyratos[3]

[1] Clermont Université, CNRS, Université Blaise Pascal, LIMOS UMR 6158
laurent.dorazio@univ-bpclermont.fr
[2] ENSEA, CNRS, Universite de Cergy Pontoise, ETIS UMR 8051
dominique.laurent@u-cergy.fr
[3] UniverSud Paris, CNRS, Université Paris Sud, LRI UMR 8623
nicolas.spyratos@lri.fr

Abstract. Cloud computing provides access to "infinite" storage and computing resources, offering promising perspectives for many applications (medicine, nuclear physics, meteorology, etc.). However, this new paradigm requires rethinking of database management principles in order to allow deployment on scalable and easy to access infrastructures, applying a pay-as-you-go model. This position paper introduces building blocks to provide cost-aware semantic caching. To this end, we first introduce cost models for data management in the cloud, then we present a semantic caching framework providing finely tuned caches for different data analysis systems. This semantic caching framework is then discussed in the context of our previous work on rewriting rules and cache management for OLAP queries. Finally, it discusses the problem of query evaluation in the cloud in presence of semantic caches as a multi-criteria optimization problem.

Keywords: Databases, Data warehouses, Cloud computing, Cost models, Caching, Optimization.

1 Introduction

Cloud computing [6] aims to tackle increasing needs of computing and storage resources, enabling to envision data management at an unexpected scale in various contexts (medical imagery, particles physics, cultural heritage). As a consequence, it has recently attracted an increasing interest, in particular by major IT companies such as Google, Microsoft, Amazon, Yahoo! or Facebook. These providers offer different data management systems in the cloud using various pricing models, like large scale systems with a simplified query interface (like Amazon SimpleDB [4] or DynamoDB [1]), or fully relational but less scalable systems (such as Amazon RDS [3] and SQL Azure [22]). In addition, several data intensive analysis tools have been proposed, such as Pig [24], Hive [27], SCOPE [8] or Jaql [7] differing in their data model and querying language.

Performance of these systems usually relies on brute force (i.e. using more numerous and/or powerful nodes) resulting in high cost and suboptimal resource

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 21–30, 2013.

management. Performance optimization in databases has been studied for years, in particular using methods such as indexing, materialized views, prefetching or caching. These methods would help improve the performance in the cloud, optimizing resources management. In particular, semantic caches [18,11] enable to rewrite queries so as to reuse local results from previous requests. Nevertheless, a cache is efficient only if it is tuned for a given context. The combination of various data management systems and pricing models lead to a major challenge: how to select relevant caching strategies according to a given provider, a specific data management system and some constraints on performance and/or budget.

Some recent approaches [12,17,28] address part of the problem, considering data sharing and optimization with respect to the pay-as-you-go model. These solutions are quite orthogonal to ours, since we consider a specific type of cache, namely semantic caching, studying in details the query evaluation process. In a previous work, we have proposed CoopSC [29] consisting of semantic caches relying on a P2P system; our preliminary experiments on CoopSC in the cloud highlighted potential savings. However, CoopSC caching strategies do not consider elasticity of the cloud with regard to the budget, nor do they take into consideration the data model or language of the different systems. Recently, rewriting rules were integrated in SCOPE, so as to consider common subexpressions [25]. Nevertheless, this approach is system dependent and does not take into account monetary aspects.

In this paper, we first address the problem of semantic caching with various data management systems, and then the query evaluation problem in the presence of cache as a multi-criteria optimization problem. To achieve this goal, our main contributions are (1) the design of generic cost models for cloud providers with regard to data management, (2) a semantic cache framework for cloud data analysis systems, (3) rewriting rules for maximizing a cache utility for OLAP queries and (4) a cost-aware cache.

The remainder of this paper is organized as follows. In Section 2, we provide the background notions used throughout the paper, and in Section 3, we present the different contributions to provide a cost-aware semantic cache in the cloud. In Section 4, we discuss the state of the art and compare it to our approach, and in Section 5, we conclude the paper.

2 Background

2.1 Cloud Computing

Cloud providers like Amazon, Google, or Microsoft, supply a pool of resources, such as hardware (CPU, storage, networks), development platforms or services. Each provider offers different services and pricing. In order for the reader to have an overview of the applied prices, the following example presents Amazon Web Services's (AWS) offer.

In AWS, Elastic Compute Cloud (EC2) provides computing resources. Different instances can be rent (small, large, very large etc.) at various prices.

For example, the cost for a small instance (1.7GB RAM, 1 EC2 Compute Unit, 160GB of local storage with a LINUX operating system) is $0.12/hour.

In AWS, input data transfers are free, whereas output data transfers vary according to the amount of data. Indeed for the first GB of data, the user will be free of charge, whereas for the next 10 TB (s)he will pay 0.12 $ per month, then 0.09 $ per month for the next the 40 TB and so on.

Amazon Elastic Block Store (EBS) is another service supplying storage capabilities. In Amazon, users pay per month according to the amount of data they store. The price is 0.10 per GB and per month.

2.2 Semantic Caching

Semantic caching [18,11] manages the content of the cache as a set of query results, usually called semantic regions. Figure 1 illustrates query processing by a semantic cache. When a query is posed, it is decomposed into two disjoint parts: the probe query retrieving part of the result already present in the cache and the remainder query which is used, if it is not empty, to retrieve missing objects.

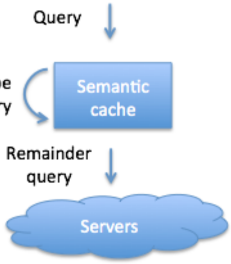


Fig. 1. Costs involved in cloud data management

Example 1. *Let us consider a semantic cache containing a semantic region associated to the predicate year=2012 and a posed query author=d'Orazio. In that case, the probe query is year=2012 AND author=d'Orazio and the remainder query is author=d'Orazio AND year!=2012.*

Semantic caching helps to reduce both the bandwidth consumption and the load on servers and thus usually decreases the response time. However, the management of the semantic cache is more complex than with a basic one. That is why such a solution is relevant in specific cases, where queries are semantically related.

2.3 Motivation

Infinite resources provided by cloud computing lead to new way to consider semantic caching. Indeed, semantic caching may sometime help to save money, in particular reducing bandwidth consumption or avoiding some computations on servers. However, it may also lead to longer response times, in particular when queries are processed on a large number of nodes relying on the elasticy property of cloud computing.

In this paper, we first address the problem of semantic caching with various data management systems and then the query evaluation in presence of cache as a multi-criteria optimization problem. To achieve this goal, our main contributions are (1) the design of generic cost models for cloud providers with regard to data management, (2) a semantic cache framework for cloud data analysis systems, (3) rewriting rules for maximizing a cache utility in presence of OLAP queries and (4) a cost-aware cache.

3 Cost-Aware Semantic Caching for Cloud Computing

3.1 Cost Models for Data Management in the Cloud

Our preliminary prior work provided cost models for materialized views in the cloud [23]. It relied on a simplified pricing model, inspired from AWS's offer. Let C_c be the sum of computing costs, C_s be the sum of storage costs and C_t be the sum of data transfer costs. Then, the total cost C for cloud data management in this simplified model is:

$$C = C_c + C_s + C_t. \tag{1}$$

Data transfer cost depends on several parameters: the size of the data set D, the amount of data related to queries Q and query results R, and the pricing model Pm applied by the CSP. Data transfer costs can be expressed by the following function:

$$C_t(D, Q, R, Pm) \tag{2}$$

It has to be noted that in order to propose a general cost model, it is mandatory to take into account both in and out transfers. However, some providers do not charge in transfers. In that case, Q and D can be simplified.

Computing cost depends on the connection's length T to process the workload Q, the type (micro, small, medium and so on) and number of nodes to be used IC and the pricing model applied by the CSP Pm. Computing cost can thus be expressed by the following function:

$$C_c(T(Q), IC, Pm) \tag{3}$$

Storage cost depends on parameters such as the CSP's pricing policy Pm, the size of the data D, the storage time T and the computing instances IC. Storage cost can then be expressed by the following function:

$$C_s(T, D, Pm, IC) \tag{4}$$

In order to propose a generic cost-aware semantic cache, we aim at extending this work, proposing a cost model for data management in the cloud, taking into account the pricing model of the main providers such as Amazon, Microsoft and Google and extending it with their specificities (licences, internal data transfers, etc.).

3.2 Semantic Caching Framework

In a previous project, we proposed ACS [14] a cache framework enabling to implement several caches to be finely tuned and to be used in grid computing. In many aspects, ACS can easily be reused in cloud computing.

ACS captures elements common to all types of cache. It distinguishes basic features such as management and replacement. Management refers to operations for managing the cache entries: adding, updating, removing and looking for an element. Replacement is in charge of choosing elements to be removed from the cache, according to a chosen replacement policy (for example FIFO, LRU, LFU).

```
(1,’dOrazio’,{(’CNRS’,6158)(’Clermont’,’UBP’)},[’dpt’
→ 63])
(2,’Laurent’,{(’CNRS’,8051)(’Cergy’,’UdC’)},[’dpt’
→ 95])
...
```

(a) Pig

```
[
 {
   name: "dOrazio",
   affiliation:
   {
     institute: "CNRS",
     number: 6158,
     university: "Clermont",
     department: "UBP"
   }
   zip: 63
 },
   name: "Laurent",
   affiliation:
   {
     institute: "CNRS",
     number: 6158,
     university: "Cergy",
     department: "UdC"
   }
   zip: 95
 } ,
 ...
]
```

(b) Jaql

Fig. 2. Examples of data

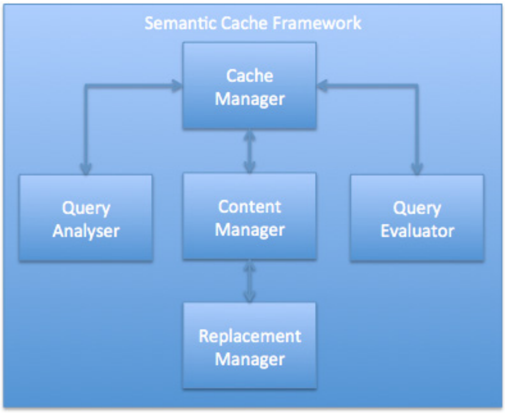


Fig. 3. The Semantic Cache Framework Architecture

The framework relies on the concept of cache entry to consider any data format, making it possible to adapt our solution to the different cloud data analysis systems. Figure 2 presents two disctinct models, JSON used in Jaql and a specific format employed in Pig.

This work will extend ACS to propose a Semantic Caching Framework (see Figure 3). It will enable to create semantic caches and provide high flexibility for deploying a cache architecture. The proposed framework will define two distinct, but cooperative, functionalities namely query analysis and query evaluation. Query analysis consists in the semantic process of comparing submitted query with the content of the cache to deduce semantic overlap or mismatch. In particular it identifies one or several cache entries to be reused to answer a posed query. Query evaluation corresponds to a set of operators (selection, projection, order, group by, etc.) to locally evaluate queries on objects stored in the cache.

Some caching mechanisms have been designed for cloud computing, such as ElastiCache [2] or Memcached [21]. They consider spatio-temporal constraints using techniques for load balancing and dynamic data placement. These approaches are complementary to our solution, which focuses on semantics and the trade-off between a price cost and performance.

3.3 Rewriting Techniques for OLAP Queries

A common approach to optimizing query evaluation in a data warehouse is to pre-compute several queries, store the answers in a cache and then reuse these answers in the query evaluation process. These pre-computed queries are commonly referred to as *materialized views* and the problem of evaluating a query by using one or more of these pre-computed results is known as the problem of answering queries using views.

This process however requires three problems to be solved: (a) allow as many queries as possible to be rewritten in terms of queries from the cache, (b)

provide a rewriting method for these queries, and (c) devise a method for storing mutually non redundant queries in the cache (*i.e.,* optimize cache containment).

In [19], we have proposed novel solutions to these problems, in a common formalism. Our approach is based on the partition semantics introduced in [26]. More precisely we associate every OLAP query Q with a partition $\Pi(Q)$, whose blocks are the sets of tuples that are grouped together by the `GROUP BY` clause in Q when computing aggregates. It has been shown in [19] that, given a query Q, the set of all queries Q' associated with the same partition as Q can be easily characterized using the functional dependencies that hold on the database. Moreover, partitions also allow to compare queries, and we show that such comparisons can be expressed using functional dependencies.

Using these theoretical results, in our approach, and contrary to all other approaches, in order to rewrite as many queries as possible using a given query Q, we do *not* store the answer to Q, but we augment this answer by additional attributes of two kinds: (i) attributes occurring in the closure (under the functional dependencies of the database) of the attribute set of the answer to Q, and (ii) attributes storing aggregate values not occurring in Q. In doing so, we address the problem (a) mentioned above, while showing that the extra storage of these attributes is not significant. Moreover, we show that, based on our query comparison, problem (b) above can be easily answered, that is, we do provide an actual rewriting method, when this is possible. Regarding problem (c) mentioned above, we show that cache optimization can be achieved using the *same* formalism of partition semantics as for query optimization. Namely we show that storing comparable queries is redundant, thus providing an effective way for optimizing the content of the cache. In particular, rewriting techniques can be used to efficiently manage entries in a semantic cache.

Based on this work, we aim at studying the evaluation of OLAP queries by considering the different query languages available in the cloud, like Pig Latin, HiveQL, SCOPE or Jaql. We will then extend the rewriting rules to take into account the specificities of these languages. Query rewriting is a complex task. In order to solve part of the problem we will ocus on a subset of queries that are frequently used, that is to say OLAP query.

We recall in this respect that the main goal of this work is to supply cost-aware semantic caching to fit cloud computing. Relying on the cost models, the semantic cache framework and the rewriting rules we will use multi-criteria optimization to enable users specifying objectives: (1) minimizing the response time given a fixed budget, (2) minimizing the cost given a deadline or (3) finding a trade-off between time and cost.

4 Related Work

4.1 Rewriting

The problem of optimizing semantically related queries has received a lot of attention during the last decades. Main references are presented into a wide survey on this topic [16].

Our approach differs from these solutions in two main ways. First, to the best of our knowledge, because [19] is the first approach to OLAP query rewriting in which the content of the cache is optimized. Second, our approach is meant to take into account specificities of NoSQL systems (considering not only the well known relational model) and their associated query languages.

4.2 Semantic Caching

Semantic caching has been studied in several contexts: distributed databases [18,11], web [9,10,20] and grid computing [15]. These solutions differ in several ways. The cache can thus store query results [11] called *semantic regions* or *semantic segments*, objects to be strongly [18] or independently [15] associated to and possibly shared by predicates. Some of them focus on a specific data structure like XML [9], [20]. Efficient research, via *signature files* has been proposed for keyword based and conjunctive queries. In a previous work, we deployed P2P semantic caches, called CoopSC [29] to highlight potential money savings in the cloud, in addition to traditional time and bandwidth consumption reductions.

All these techniques are complementary to our project and can be reused to provide finely tuned caches. However, none of them takes into account properties such as elasticity and pay-as-you-go.

4.3 Data Management in the Cloud

Data management in the cloud has attracted a lot of attention during the last years. Major IT companies like Amazon, Facebook, Google, IBM, Microsoft or Yahoo! have provided large scale data management systems. Some of them consists on large scale solutions providing a simplified query interface (usually a subset of SQL), like Amazon SimpleDB [4] or DynamoDB [1]. Less scalable but fully relational approach are also available, such as Amazon RDS [3] and SQL Azure [22]. Data intensive analysis tools such as Pig [24], Hive [27], SCOPE [8] or Jaql [7], relying on massively parallel execution environment like MapReduce [13], or its open source version Hadoop [5] have been developped.

To the best of our knowledge, caching has not been intensively addressed in these systems (even if some works start to consider reusing common subexpressions for query processing [25]). In particular none of them takes into account semantic caching and optimization strategies with regard to elasticity and pay-as-you-go. Our solution will consider these aspects and provide strategies that can be used with any of these systems.

4.4 Cost-Aware Data Management

The pay-as-you-go model leads to consider data management in general, and optimization particularly, from a novel point of view. Recent approaches [12,17,28] have addressed the challenge of selecting the optimizations to implement and the way to price them in a data shared environment. A first solution [12,17] consists in asking users their willingness to pay for some optimizations, monitoring the workload and choosing the techniques to implement based on solutions that would have been useful in the past. Unfortunately, this solution has two main limitations: (1) it assumes that users are honest and (2) it does not guarantee that users will recover the cost of an optimization. These issues can be addressed using Mechanisms Design [28], an area of game theory, which enables to provide solutions to an optimization problem in presence of untrusted players.

These solutions studied the optimization in a shared environment at a higher level than we do. Our proposal will focus on a particular technique, that is to say semantic caching, trying to address optimization individually (from the client point of view) and dynamically.

5 Conclusion

In this position paper, we presented building blocks to provide cost-aware semantic caching for cloud computing. We first introduced cost models for data management in cloud computing. We then described a semantic caching framework to consider the different data analysis systems. After that we introduced some rewriting rules to maximize caches utilization. Finally we considered the optimization process with regard to a given user's objective. We are currently implementing and validating these contributions.

References

1. Amazon. Dynamodb. Web page, `http://aws.amazon.com/dynamodb/`
2. Amazon. Elasticache. Web page, `http://aws.amazon.com/elasticache/`
3. Amazon. Rds. Web page, `http://aws.amazon.com/rds/`
4. Amazon. Simpledb. Web page, `http://aws.amazon.com/simpledb/`
5. Apache. Hadoop. Web page, `http://hadoop.apache.org/`
6. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)
7. Beyer, K.S., Ercegovac, V., Gemulla, R., Balmin, A., Eltabakh, M.Y., Kanne, C.-C., Özcan, F., Shekita, E.J.: Jaql: A scripting language for large scale semistructured data analysis. PVLDB 4(12), 1272–1283 (2011)
8. Chaiken, R., Jenkins, B., Larson, P.-Å., Ramsey, B., Shakib, D., Weaver, S., Zhou, J.: Scope: easy and efficient parallel processing of massive data sets. PVLDB 1(2), 1265–1276 (2008)
9. Chen, L., Rundensteiner, E.A., Wang, S.: Xcache: a semantic caching system for xml queries. In: SIGMOD, Madison, Wisconsin, USA, p. 618 (2002)
10. Chidlovskii, B., Borghoff, U.M.: Semantic caching of web queries. VLDBJ 9(1), 2–17 (2000)
11. Dar, S., Franklin, M.J., Jonsson, B.T., Srivastava, D., Tan, M.: Semantic data caching and replacement. In: VLDB, Bombay, India, pp. 330–341 (1996)
12. Dash, D., Kantere, V., Ailamaki, A.: An economic model for self-tuned cloud caching. In: ICDE, Shanghai, China, pp. 1687–1693 (2009)
13. Dean, J., Ghemawat, S.: Mapreduce: Simplified data processing on large clusters. In: OSDI, San Francisco, California, USA, pp. 137–150 (2004)

14. d'Orazio, L., Roncancio, C., Labbé, C.: Adaptable cache service and application to grid caching. Concurrency and Computation: Practice and Experience 22(9), 1118–1137 (2010)
15. d'Orazio, L., Traore, M.K.: Semantic cache for pervasive grids. In: IDEAS, Cetraro, Italy, pp. 227–233 (2009)
16. Halevy, A.Y.: Answering queries using views: A survey. VLDBJ 10(4), 270–294 (2001)
17. Kantere, V., Dash, D., Gratsias, G., Ailamaki, A.: Predicting cost amortization for query services. In: SIGMOD, Athens, Greece, pp. 325–336 (2011)
18. Keller, A.M., Basu, J.: A predicate-based caching scheme for client-server database architectures. VLDBJ 5(1), 35–47 (1996)
19. Laurent, D., Spyratos, N.: Rewriting aggregate queries using functional dependencies. In: MEDES, San Francisco, CA, USA, pp. 40–47 (2011)
20. Lillis, K., Pitoura, E.: Cooperative xpath caching. In: SIGMOD, Vancouver, BC, Canada, pp. 327–338 (2008)
21. Memcached. Memcached. Web page, `http://memcached.org/`
22. Microsoft. Sql azure. Web page, `http://www.windowsazure.com/en-us/home/features/data-management/`
23. Nguyen, T.-V.-A., Bimonte, S., d'Orazio, L., Darmont, J.: Cost models for view materialization in the cloud. In: DanaC@EDBT, Berlin, Germany, pp. 47–54 (2012)
24. Olston, C., Reed, B., Srivastava, U., Kumar, R., Tomkins, A.: Pig latin: a not-so-foreign language for data processing. In: SIGMOD, Vancouver, BC, Canada, pp. 1099–1110 (2008)
25. Silva, Y.N., Larson, P.-A., Zhou, J.: Exploiting common subexpressions for cloud query processing. In: ICDE, Washington, DC, USA, pp. 1337–1348 (2012)
26. Spyratos, N.: The partition model: A deductive database model. ACM TODS 12, 1–37 (1987)
27. Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Anthony, S., Liu, H., Murthy, R.: Hive - a petabyte scale data warehouse using hadoop. In: ICDE, Long Beach, California, USA, pp. 996–1005 (2010)
28. Upadhyaya, P., Balazinska, M., Suciu, D.: How to price shared optimizations in the cloud. PVLDB 5(6), 562–573 (2012)
29. Vancea, A., Machado, G.S., d'Orazio, L., Stiller, B.: Cooperative database caching within cloud environments. In: Sadre, R., Novotný, J., Čeleda, P., Waldburger, M., Stiller, B. (eds.) AIMS 2012. LNCS, vol. 7279, pp. 14–25. Springer, Heidelberg (2012)

Specifying the Federation Structure among Application Smart Objects by Example through Direct Manipulations

Jeremie Julia and Yuzuru Tanaka

Meme Media Laboratory,
Hokkaido University, Sapporo, Japan
{jeremie.julia,tanaka}@meme.hokudai.ac.jp

Abstract. This paper focuses on the final stage of a new approach to manage automatic federations of smart objects (SOs) or smart mobile devices. This new approach models each smart object federation as a catalytic reaction. Each reaction is modeled as an RNA (RiboNucleic Acid) replication with or without a regulation switch in the biological world. Thus, it is possible to describe a complex scenario as a catalytic reaction network where the result of a reaction, i.e. the result of a federation, may work as a source material for another reaction or as a catalyst to enable or disable another reaction. The SOs that we want to federate are called application smart objects. To each application SO, we attach an NSO (Nucleotide SO) of its corresponding type as its tag. The reaction process federate those tag NSOs, but not the application SOs they are attached to. Once the NSOs are federated, the context of the reaction can send them a program to connect the application SOs directly with each other. In this paper, we propose a solution to generate the rules composing such a program. We also briefly present a proof of concept of our system through its implementation with Sun SPOT devices.

Keywords: Human-computer Interaction, Ubiquitous Computing, Pervasive Computing, Service Federation, Smart Object.

1 Introduction

It is now common to have our environment filled with smart objects (SOs), including smart phones, PDAs, tablet PCs, embedded computers, sensor devices and RFID tags[1,2]. However, they generally stay disconnected from each other or their connections do not dynamically change. Moreover, in the research and development of ubiquitous computing, we are confronted with two stereotyped scenarios: location-transparent service continuation, and location- and/or situation-aware service provision. We believe that by allowing them to dynamically federate with each other, we could use their full potentiality. It was shown in [3,4] that this situation comes from the lack of formal models. Other researchers also think that the potentiality of the ubiquitious computing is limited by the lack of formal modeling[5,6]. Thus, to consider scenarios beyond these

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 31–40, 2013.

two stereotyped ones, it is necessary to provide a new formal model allowing the description of complex smart object federation scenarios. Milner thought that a single model can not handle all the concepts needed to understand ubiquitous computing, and so, a hierarchy of models is necessary. He called this hierarchy "the tower of models" [7,8], in which each higher model should be represented or implemented by a lower model.

The formal models proposed by Tanaka [1], organised in a tower of models, deal with the following three different levels:

- The first level: The port matching model is used to describe federation (connections) and interoperation mechanisms.
- The second level: Graph rewriting rules are used to describe dynamic change of federation structures.
- The third level: The catalytic reaction network modeling is used to describe complex application scenarios with mutually related federations.

A catalytic reaction network is a set of reactions. The output material of a reaction may work as the input material of another reaction, or as a catalyst to enable or disable another reaction. We use this system to model complex federation scenarios of SO, where the materials denote SOs. The SOs we want to federate are the SOs that are used to construct applications, and thus are called application smart objects. There are also meta SOs, that are necessary to implement a reaction, but that are not the target SOs that we want to federate.

Each reaction is modeled as an RNA replication process. To perform such a replication, we introduced the nucleotide smart objects (NSOs), that are meta SOs. They are objects that behave like the nucleotides in the RNA world hypothesis. Their behavior is described by graph rewriting rules in the second layer modeling. For each application SO, an NSO corresponding to its type is attached to it as its tag. As the result of the reaction, the tag NSOs are federated.

What is still missing in the previous research is the mechanism for the connections among the application SOs after their tag NSOs become federated. In this paper, we will propose a solution by using instructor SOs, that are also meta SOs. These meta SOs are used to generate the connection rules that the NSOs have to know to connect the application SOs.

In the next section, we will define what a catalytic reaction network denotes and how to design a scenario based on a catalytic reaction network in our formal models. In the following section, we will briefly review the definition of smart objects. After that, we will define nucleotide smart objects, that are used to implement a catalytic reaction network. Then, we will show how to define the connection among the smart objects involved in a reaction, using instructor smart objects. Finally, we will show our implementation of the nucleotide smart object using SunSpot modules.

2 Catalytic Reaction Network

As explained before, catalytic reaction network is a set of reactions in which the result of a reaction may work as an input material or as a stimulus of another

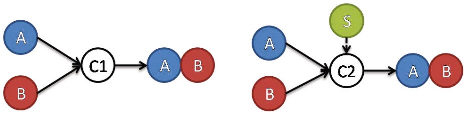


Fig. 1. Example Reactions

reaction. Fig 1 presents two reactions. The input materials of both reactions is A and B, and the output materials of both are AB. But one of them needs a stimulus S to happen. A context, which is also a catalyst, is associated with each reaction. The contexts in Fig 1 are C1 and C2. In order to activate a reaction, the input materials and the stimulus of the reaction have to enter the context of the reaction. In our model, the input/output materials and stimuli are application SOs, and each federation is modeled as a reaction. The contexts are composed of meta SOs, and the scope of a context is the wireless range of the SOs constituting it. We say that the input material and stimulus enter a context when they enter the scope of the context.

Fig. 2 shows an example scenario with more than one federation. This scenario was modeled as a reaction network. The application SOs are represented by colored circles. The context of each reaction is represented by a white circle. This scenario is about a museum providing more than one exhibition in parallel to its visitors. Each exhibition exposes many exhibits. The museum may offer more than one visitor-activity plan to its visitors. Each visitor-activity plan is a sequence of activities allowing each visitor to interact with the exhibits. As a part of our scenario, the museum provides two different visitor-activity plans: a guided tour, and an augmented reality tour. For both of these plans, the visitor first needs to federate his PDA P with his headphone H to get the federated SO P-H. This will be done automatically when he passes through the first gate G1 with P and H. If he does not have one (or any) of these objects, he can pick up one (or both) at the entrance of the museum. Later, when he enters the waiting room for a guided tour, the gate G2 will federate his already federated object P-H with the tablet G of the guide. Then, during the tour, the guide sends audio, video and text information to the headphone and to the PDA of the visitor. The guide will choose the information he sends depending on what to present to the visitor. The visitor can also participate in an augmented reality tour. The visitor needs to register for the tour, because the number of virtual reality (VR) glasses provided by the museum for a tour is limited. When he registers, he gets a ticket. This ticket controls access to the tour, and works as a stimulus to connect visitor's PDA-headphone to the VR glasses when he passes through the gate G3 to enter the first room of the tour. Then, the visitor will be able to control the VR with the PDA and the VR glasses will be able to send sound directly to the headphones. If someone tries to join the tour without a ticket, they will not be able to use the VR glasses. In this scenario, all the connections and federations are done automatically by passing through some gate, and no visitor operation is necessary.

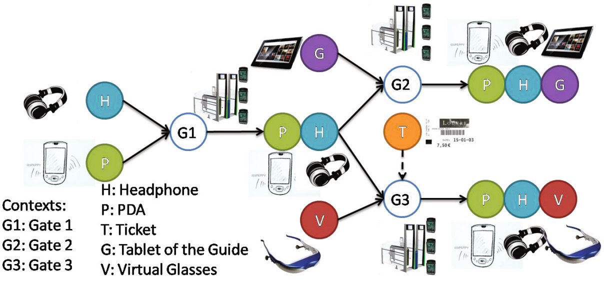


Fig. 2. An example catalytic reaction network representing a scenario of smart-object federations

3 Smart Objects

In this section, we briefly review the definition of smart objects (SOs). More information can be found in [2]. A SO has five properties: its identifier, its set of types, its state, its set of ports, and its set of rules. Each port of a SO has its polarity, depending on whether the port is a service requesting port ('-' polarity) or a service providing port ('+' polarity). Each port also has its type. Thus, a providing (or requesting) port having the type t will be denoted as +t (or -t). A t port of an object A can establish a connection to a t port of another object B if they have the same type and a different polarities. Thus a connection will be denoted as a channel t from A to B. A port can have only one active connection. Moreover, each port has its state, which is either hidden or visible. Each port state determines which kind of connection the port can establish. In a federation of SOs, we can refer to an object C from another object A if we have a path σ from A to C. A path σ is a sequence of connections in the federation separated by a dot. For example, if we have a connection t from A to B, and a connection u from B to C, then A can refer to C using the path $\sigma = t.u$. We say that C is the target of σ, and we call C the σ object of A.

4 Nucleotide Smart Objects

In this section, we will briefly review the nucleotide smart objects (NSOs) introduced in [1]. More information can be found in [2]. The NSOs are the first meta SOs presented in this paper. They have a similar behavior as the nucleotide in the RNA world. We use this behavior to make a reaction. Each NSO is an SO that has five ports. Two L ports (a requesting one and a providing one) and a requesting P port independent on the type of the NSO, and two B ports (a requesting one and a providing one) dependent on the type of the NSO. As shown in Fig. 3, an NSO can be attached through its P port to an application SO. The type of the NSO depends on the type of the application SO to which it can be attached. Thus, the type of an NSO is noted as N_x, with x representing the type of the application SO. For example,N_T is the type of an NSO that can

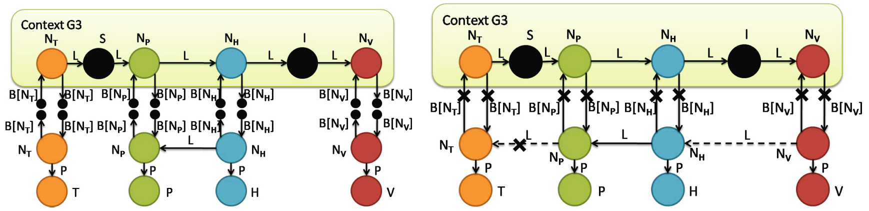


Fig. 3. An implementation of an example reaction with NSOs. The left diagram shows the initial situation of the reaction. The right diagram shows the situation after the docking step, the output material formation, and the undocking step

only be attached to an application SO of type T. Such an NSO can also be used to compose a context (the context G3 in this example) by linking it to another NSO through their L ports. An NSO attached to an application SO is called an NSO tag. There is another kind of NSOs called separators that are used only in the context. There are two different separators, a stimulus separator S, and an input material separator I, they are represented by black circles in the example figures. Because they are not used as NSO tags, they do not have B ports, nor P ports. They are used to indicate which parts of the context corresponds to the stimulus, and input material. The NSOs between two I, or between an I and an S or an I and the right end, correspond to input material parts. In the example, N_P and N_H are between an S and an I, this means that the reaction corresponding to the context G3 accepts an already federated object P-H as an input material. The NSOs between the left end and S define the stimulus part. In the example, the stimulus is a single SO T, but we could have a composite SO as a stimulus. The initial state of the reaction is represented by the left diagram in Fig. 3. There are the context G3 with the application SOs T, P-H, and V, all attached to their NSO tags. The reaction process consists of three main steps: the docking step, the output material formation step, and the undocking step. Those are all presented in the right diagram in Fig. 3. The docking step will connect all the NSO tags with the context NSOs through B ports, and we will finally obtain, the situation in Fig. 3 without the dotted arrows. Then, the second step will establish the L connections, corresponding to the dotted arrows in the diagram, through the input NSOs. And then, the undocking step will disconnect all the connections with cross in the diagram, i.e. the L connection between N_T and N_P, and all the B connections. After the reaction, because N_V can reach V, H and P, some connection program downloaded from the context can establish arbitrary connections among application SOs P, H and V.

5 Specifying the Connections between the Smart Objects

In the previous section we showed how, after a reaction, at least one tag NSO can reach all the application SOs. We still need to directly connect the application SOs. For this purpose, each program, called connection program, stored in each

NSO of the context may be sent to the tag NSOs. Such a program will tell the tag NSOs how to connect the application SOs with each other. By extending direct manipulation for defining contexts proposed in [2], we like to automatically generate these programs only through direct manipulations of SOs. We will use special generic SOs, called instructors, for specifying the connection programs and send them to the NSOs of the context. Instructors are the second meta SOs in this paper. They will extract connection rules from user's direct manipulation operations. They have two L ports and two B ports like the NSOs, and they should be compatible with the NSOs for the connections B. They provide a UI allowing the users to change their types (to corresponding types of the NSOs), to add custom ports and make manual connections. Fig. 4 shows an example of using the instructors to create the connection programs and send it to the NSOs of the context G3. In the step 1, by using the UI of instructors, we instantiate their types to the corresponding types of the different NSOs of G3. In the step 2, we connect manually the instructor N_H to the instructor N_P through their L ports. This connection is necessary because the context G3 is expected as an input material an N_H and an N_P already federated. Then, we can create custom ports, and establish connections manually between the instructors by using those custom ports. Those connections compose a federation example that we want to replicate among the application SOs. In our scenario example, we want the following connections among the application SOs:

- A sound connection from the VR glasses to the headphone to send sound.
- A UI connection from the VR glasses to the PDA to send the UI that the PDA displays for a user to control the VR glasses.
- A control connection from the PDA to the VR glasses used to control the VR glasses.

That is why we created those connections among the instructors representing the application SOs. Once we build all the desired connections, the instructors are put in the scope of the context in which the connection programs will be stored. In our example, this context is G3. Because the instructors are compatible with the NSOs, the reaction in G3 will start, and we will reach the step 3 in Fig 4. Now all the instructors are linked through the L connections, then they will

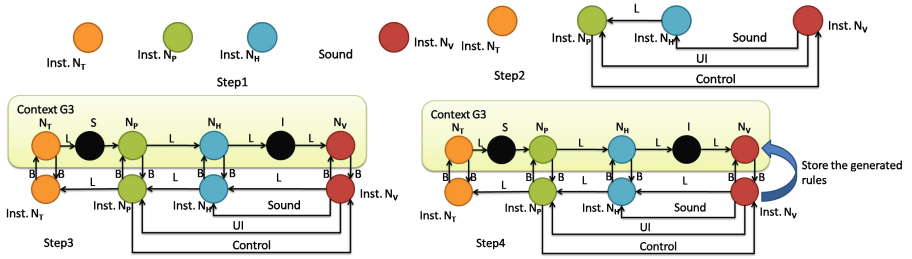


Fig. 4. An example showing how to design the connections among application Smart Objects using the instructor smart objects

invoke the rule extraction routine to generate the connection programs. The rule extraction routine generates connection rules using the following two rules.

Let L^n be a path of n connections of type L, where $n > 0$.

First rule: if an instructor I can reach another instructor J through an L^n path; and if I can also reach J through a direct connection C that is not a L connection; then generate the rule "span a channel C from the SO at $\sigma = P$ to the SO at $\sigma = L^n.P$".

Second rule: if an instructor I can reach another instructor J through an L^n path; and if J can also reach I through a direct connection C that is not a L connection; then generate the rule "span a channel C from the SO at $\sigma = L^n.P$ to the SO at $\sigma = P$".

The set of rules generated by an instructor compose one connection program. For example, the instructor N_V will generate three rules corresponding to the three connections we designed: "span a channel Sound from the SO at $\sigma = P$ to the SO at $\sigma = L.P$", "span a channel UI from the SO at $\sigma = P$ to the SO at $\sigma = L.L.P$" and "span a channel Control from the SO at $\sigma = L.L.P$ to the SO at $\sigma = P$". The connection program consisting of these three rules is stored into the NSO N_P of the context during the step 4. Finally, when a reaction happens, the NSOs of the context send their connection programs to the tag NSOs. Each tag NSOs will execute the received connection program to replicate the connections among the application SOs. These replicated connections are the same as those instructed by the user among the instructors during the step 2. In the Fig 5, those connections are shown as dashed arrows.

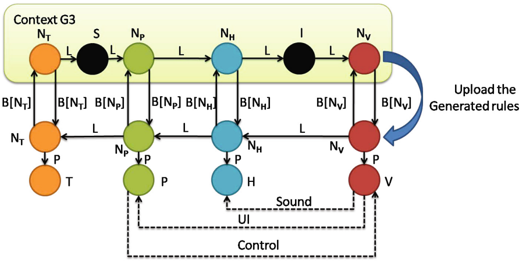


Fig. 5. Execution of the generated connection software

6 Implementation with Sun SPOT

We have implemented our solution using Sun SPOT[1] modules as a proof of concept. The Sun SPOTs are small devices that use IEEE 802.15.4 standard for wireless communication, sense the environment, and display information with LEDs. They integrate a JVM and thus are programmable with Java, which makes them one of the easiest SO devices for prototyping.

Fig. 6 shows the architecture of our implementation. Our architecture is divided into three layers. The first layer, the base smart object layer, provides all

[1] `http://www.sunspotworld.com/`

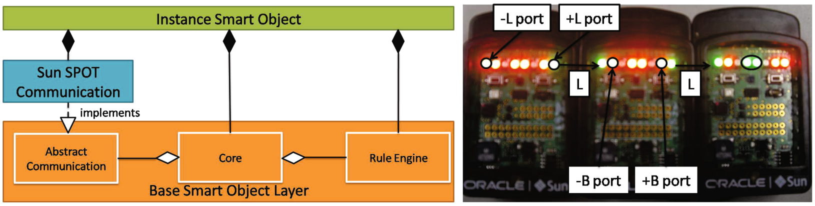


Fig. 6. The left diagram shows our smart object architecture, while the right picture shows an example context made of the three Sun SPOTs

the features common to all the SOs, i.e. the core module that manages all the properties of the SO (state, type, oid, ports, etc. . .), the abstract communication module that manages the message exchange among different SOs, and the rule engine module that executes the set of rules. The core needs to collaborate with an instance of both the communication module and the rule engine module for its execution. This layer is independent on the device, that is why the communication module is abstract. Some components of this module are independent from the device. An example of such components is the one to treat the received messages. The other components depending on the device have to be implemented in another layer. It is the role of the second layer to implements those components. In our case, the second layer implements all the components needed by the first layer to make wireless communications using the Sun SPOTs device communication capacities. The third one, the instance layer, corresponds to the instantiation and wrapping of the core, the rule engine, and the communication modules. It is in this layer that the developer defines the SO properties (type, ports, rules, . . .). Because only the second layer depends on the device, it is easy to reuse the code of a SO for another device (if this device has a JVM). It is done by changing only one line of code in the instance layer to refer to another communication module, implementing the communication for this new device. Thus, the program that we developed for the Sun SPOT can easily be reused for devices using Android[2] for example. Moreover, the core provides a system of observer/observable that allows the instance layer to receive events when something changes in the core and to display appropriate information. Such an event is for example the state change of a port. We use the LEDs for displaying these information with the Sun SPOT (see Fig. 6).

The following conditions and actions are implemented in the rule engine module. They allow the developer to implement all the primitive conditions and actions used in the description of graph rewriting rules[2].

Primitive conditions:

- HasState(σ, state): to check the state of the σ object.
- HasType(σ, type): to check the type of the σ object.
- InScope($\sigma 1$, $\sigma 2$): to check if the $\sigma 2$ object is in the scope of the $\sigma 1$ object (the scope of an object it is wireless range).

[2] http://www.android.com/

- IsConnected(σ, polarity, portType): to check if the port of the σ object with this polarity and this port type is connected.
- IsFree(σ, polarity, portType): to check if the port of the σ object with this polarity and this port type is free (which means it is available for a connection).
- IsHidden(σ, polarity, portType): to check if the port of the σ object with this polarity and this port type is hidden.
- IsIdentical($\sigma1$, $\sigma2$): check if $\sigma1$ object and the $\sigma2$ object are the same object.
- IsVisible(σ, polarity, portType): to check if the port of the σ object with this polarity and this port type is visible.
- Neighbor(σ, portType): to check if the σ object has, in its scope, an object with a visible + port having the port type "portType".

Each of these conditions return true if the σ path exists and if the condition is satisfied, if not it returns false. That means, it returns false if the σ path is broken.

Primitive actions:

- Expose(σ, polarity, portType): set the state of the port of the σ object with this polarity and this port type to visible.
- Hide(σ, polarity, portType): set the state of the port of the σ object with this polarity and this port type to hidden.
- SetState(σ, state): set the state of the σ object to the specified state.
- Span($\sigma1$, $\sigma2$, portType): span a connection from the - port having the port type "portType" of the $\sigma1$ object to the + port having the port type "portType" of the $\sigma2$ object.
- SpanToNeighbor(σ, portType): span a connection from the - port having the port type "portType" of the σ object to the + port, having the port type "portType" and the visible state, of an object in the scope of the σ object.

We can combine different actions and conditions to define rules. The following piece of code shows the simple rule 1 thus defined. This rule checks if the SO at $\sigma =$" ", i.e. the SO that executes this rule, has the state 1 (line 2), if it has a connected -L port (line 3). If all these conditions are satisfied, then it executes only one action (it may execute more). It will change the state of the object at $\sigma =$"L" (line 4). Finally, we add these rules in the instance of the rule engine (line 6) that will try periodically to execute the rules.

```
1. Rule rule1 = new Rule();
2. rule1.addCondition(new HasState("", "1"));
3. rule1.addCondition(new IsConnected("", "-", "L"));
4. rule1.addAction(new SetState("L", "1"));
5. ruleEngine.addRule(rule1);
```

7 Conclusion

In the previous paper[2], it was shown that we can use NSOs to construct a context improvisationally by direct manipulation. What was missing there was the design of the connection programs necessary to connect the applications SOs in the output federation. Based on a similar direct manipulation approach presented in the previous paper, we have shown here that it is possible to design the connection programs improvisationally only through using the instructor SOs. We can manually establish connections between the instructors. These connections define the federation that we want to set up among the application SOs. Then, the instructors can generate the connection programs from those manually established connections, and can inject these generated programs into the NSOs of the context. Thus, we do not need to use any additional computers nor tablet PCs to program a complex catalytic reaction network scenario. We have also briefly presented the result of our system implementation that helps us to test our model and application scenarios.

References

1. Tanaka, Y.: Proximity-based federation of smart objects: Liberating ubiquitous computing from stereotyped application scenarios. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010, Part I. LNCS, vol. 6276, pp. 14–30. Springer, Heidelberg (2010)
2. Julia, J., Tanaka, Y.: Improvisational construction of a context for dynamic implementation of arbitrary smart object federation scenarios. In: ICME Workshops, pp. 223–229 (2012)
3. Milner, R.: Theories for the global ubiquitous computer. In: Walukiewicz, I. (ed.) FOSSACS 2004. LNCS, vol. 2987, pp. 5–11. Springer, Heidelberg (2004)
4. Henricksen, K., Indulska, J., Rakotonirainy, A.: Modeling context information in pervasive computing systems. In: Mattern, F., Naghshineh, M. (eds.) PERVASIVE 2002. LNCS, vol. 2414, pp. 167–180. Springer, Heidelberg (2002)
5. Crowcroft, J.: Engineering global ubiquitous systems. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 366, 3833–3834 (2008)
6. Rodden, T.: Living in a ubiquitous world. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 366, 3837–3838 (2008)
7. Milner, R.: Ubiquitous computing: Shall we understand it? Comput. J. 49, 383–389 (2006)
8. Milner, R.: Scientific foundation for global computing. In: Priami, C., Cardelli, L., Emmott, S. (eds.) Transactions on Computational Systems Biology IV. LNCS (LNBI), vol. 3939, pp. 1–13. Springer, Heidelberg (2006)

Decision Making in Knowledge Integration with Dynamic Creation of Argumentation

Ken Satoh[1] and Kazuko Takahashi[2]

[1] National Institute of Informatics and Sokendai
ksatoh@nii.ac.jp
[2] School of Science&Technology, Kwansei Gakuin University
ktaka@kwansei.ac.jp

Abstract. We discuss a semantics of dynamic creation of arguments when knowledge from different agents are combined. This arises when an agent does not know the other agent's knowledge and therefore, the agent cannot predict which arguments are attacked and which counter-arguments are used in order to attack the arguments. In this paper, we provide a more general framework for such argumentation system than previous proposed framework and provide a computational method how to decide acceptability of argument by logic programming if both agents are eager to give all the arguments.

1 Introduction

Argumentation system is a hot topic in legal reasoning and in more general setting such as negotiation in multi-agent systems[Rahwan09] and knowledge integration[Bikakis10,Janjua12].

However, most of the work on argumentation is based on the assumption where complete information about argumentation is provided[Dung95] meaning that all the set of arguments and attack relations between them are known in advance. It would be appropriate for an application domain where we can see all the arguments and counter-arguments so that we can conclude the most appropriate result based on all the arguments. However, in reality, there would be another type of argumentation where relevant agents only have their own belief and they do not know other agents' belief and so they do not predict how other agents attack their own arguments.

Consider the following example where c and p are two parties and numbers attached with c and p express order of arguments. [1].

$p0$**:** "We should buy a smart phone A."
$c1$**:** "We should buy a smart phone B instead of A."
$p1$**:** "B is more expensive than A."
$c2$**:** "B is now on sale so B is cheaper than A"
$p2$**:** "B's battery does not long more than A."

[1] This is a modified version from[Okuno09] in which they use a criminal case.

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 41–50, 2013.

*c*3**:** "B's battery is renewed so that B can have a larger capacity."
*p*3**:** "Unfortunately, if we buy a new battery, B is no longer cheaper than A even if it is on sale.

This kind of argumentation would occur in examinations of witness in legal courts. In the above example, *c*2 is not firstly attacked but after the argument of *c*3 is given, *c*2 is attacked by *p*3. Since agent *p* does not know whether the argument of *p*3 is relevant to *c*2 even if he knows *p*3 beforehand, *p* could not use the counter-argument *p*3 at first. But after *c*3 is provided, *p* can attack *c*2 by pointing out the contradiction with *c*3. This phenomenon cannot be formalized in argumentation system based on complete information about arguments and so we need a new framework.

Pioneer work on this direction would be, as far as we know, APKC (Argumentation Procedure with Knowledge Change) [Okuno09,Takahashi11] where counter-arguments, which cannot be used at the starting point of argumentation since these counter-arguments are not convinced by the agent itself, are triggered by other agents' arguments. In this paper, we extend this direction to provide more general framework than APKC. The difference between their works and this work are as follows:

- We let an agent give as many counter-arguments against other agent's arguments as they like where as APKC allows only one counter-argument against one argument at one turn.
- We do not employ any specific strategy how to make counter-argument whereas APKC imposes an agent to stick to one line of arguments until no counter-argument is made, then the agent change counter-argument in the other line of arguments.

To formalize the above, we introduce *sources of arguments* which represent usable arguments. This means that even if there are potential counter-arguments against the other agent's arguments, the agent cannot use the argument if the argument is not in the source. We also introduce *derivation rule of sources* which represent dynamic addition of arguments which were not initially able to be used, but later become usable based on the other agent's new arguments and its own belief. By these mechanisms, we let agents not know whether potential arguments would be usable in the future since there are incomplete information about the other agents' behavior.

Then, we show a computational method to decide which arguments are accepted by translating argumentation framework into logic programming from the God's viewpoint under the assumption that all possible arguments will always be presented by both parties sooner or later.

2 Framework for Argumentation under Incomplete Information

Definition 1. *An* argumentation framework *is a quadruple,* $\langle Arg, Attack, Source, Derive\rangle$ *defined as follows.*

- *Arg is a pair,* $\langle Arg_P, Arg_C\rangle$ *where* Arg_P*(*Arg_C*, respectively) is a set called* an argument set *for* P*(*C*, respectively)*[2]*.*
- *Attack is a pair,* $\langle Attack_P, Attack_C\rangle$ *where* $Attack_P$*(*$Attack_C$*) is a subset of* $Arg_P \times Arg_C$*(*$Arg_C \times Arg_P$*, respectively) and called* an attack relation for P*(*C*, respectively). We say* P*(*C*, respectively)*attacks n' by n *if* $\langle n, n'\rangle \in Attack_P$*(*$Attack_C$*, respectively).*
- *Source is a pair,* $\langle Source_P, Source_C\rangle$ *where* $Source_P$ *(*$Source_C$*, respectively) is a subset of* Arg_P *(*Arg_C*, respectively) called* a source of arguments for P*(*C*, respectively).*
- *Derive is a pair,* $\langle Derive_P, Derive_C\rangle$ *where* $Derive_P$ *(*$Derive_C$*, respectively) is a set of the following rules of the form:*
  $$n \Leftarrow n_1, ...n_m$$
  where $n \in Arg_P$ *(*Arg_C*, respectively) and* $n_i \in (Arg_P \cup Arg_C)(1 \leq i \leq m)$ *called* a set of derivation rules for P*(*C*, respectively). We call n the conclusion of the rule and* n_i*'s conditions of the rule.*

We assume that there is no loop in $Attack_P \cup Attack_C$ to avoid infinite loop of arguments[3].

In the above definition, a derivation rule enables an agent to augment its own source of arguments by adding the conclusion of the derivation rule if condition part is satisfied.

We define an argumentation tree which gives a semantics of acceptance of arguments as follows.

Definition 2. *An* argumentation tree $Tr = \langle N, E\rangle$ *w.r.t. an argumentation framework* $\langle Arg, Attack, Source, Derive\rangle$ *is an in-tree*[4] *such that* $N \subset Arg_P \cup Arg_C$ *and* $E \subset Attack_P \cup Attack_C$ *and satisfies the following conditions:*

- *The root of* Tr *is* $p \in Source_P$ *called "conclusion".*
- *If* $\langle n, n'\rangle \in E$ *then either of the following holds.*
  - $n \in Source_P$ *and* $n' \in Source_C$ *and* $\langle n, n'\rangle \in Attack_P$*.*
  - $n \in Source_C$ *and* $n' \in Source_P$ *and* $\langle n, n'\rangle \in Attack_C$*.*

Let $Tr = \langle N, E\rangle$ *be an argumentation tree.* $n \in N$ *is accepted w.r.t.* Tr *if*

- *there is no edge to* n*, or*
- *there is no* n' *s.t.* $\langle n', n\rangle \in E$ *and* n' *is accepted w.r.t.* Tr*.*

Now, we can define a game called an *argumentation game* which gives a dialog between two parties. In argumentation game, agents can refer to source of arguments to produce counter-arguments.

Definition 3. *A* move *of an* argumentation game *w.r.t. argumentation tree* $Tr = \langle N, E\rangle$ *and a pair of source sets* $\langle S_P, S_C\rangle$ *is an expansion of* Tr*,* S_P *and* S_C *defined as follows.*

[2] P denotes "Pros" and C denotes "Cons".

[3] We may formalize an argumentation with loop if we follow Dung's stable extension or preferred extension. It is a future research topic.

[4] An in-tree is an directed tree in which a single node is reachable from every other one (See Fig.1).

- *P's move is a set $Move_P \subseteq Attack_P$ such that for every n such that $\langle n, n'\rangle \in Move_P$, $n \notin N$, $n \in Source_P$ and $n' \in N$. Then, a new set of nodes in a new argumentation tree N', a new set of edges in a new argumentation tree E' and a new pair of source sets $\langle S'_P, S'_C\rangle$ becomes the following.*
  - $N' = N \cup \{n | \langle n, n'\rangle \in Move_P\}$
  - $E' = E \cup Move_P$
  - $S'_P = S_P$
  - $S'_C = S_C \cup \{n | (n \Leftarrow n_1, ..., n_m) \in Derive_C \text{ and } n_i \in N'(1 \leq i \leq m)\}$
- *C's move is a set $Move_C \subseteq Attack_C$ such that for every n such that $\langle n, n'\rangle \in Move_C$, $n \notin N$, $n \in Source_C$ and $n' \in N$. Then, a new set of nodes in a new argumentation tree N', a new set of edges in a new argumentation tree E' and a new pair of source sets $\langle S'_P, S'_C\rangle$ becomes the following.*
  - $N' = N \cup \{n | \langle n, n'\rangle \in Move_C\}$
  - $E' = E \cup Move_C$
  - $S'_P = S_P \cup \{n | (n \Leftarrow n_1, ..., n_m) \in Derive_P \text{ and } n_i \in N'(1 \leq i \leq m)\}$
  - $S'_C = S_C$

If both agents give $\emptyset$ in consecutive two moves, then we say that the game is finished *and we call a final tree after a game is finished* argumentation game tree*. Let Tr be an argumentation game tree $\langle N, E\rangle$. We say that a node $n \in N$ is accepted w.r.t. the argumentation game tree Tr if n is accepted w.r.t. argumentation tree Tr.*

Note that a move can be $\emptyset$[5], and a conclusion is decided to be accepted or not using the argumentation game tree.

Example 1. Consider the example discussed at Introduction. Then,
$Arg_P = \{p0, p1, p2, p3\}$ and $Arg_C = \{c1, c2, c3\}$,
$Attack_P = \{\langle p1, c1\rangle, \langle p2, c1\rangle, \langle p3, c2\rangle\}$,
$Source_P = \{p0, p1, p2\}$,
$Derive_P = \emptyset$
$Attack_C = \{\langle c1, p0\rangle, \langle c2, p1\rangle, \langle c3, p2\rangle\}$,
$Source_C = \{c1, c2, c3\}$,
$Derive_C = \{p3 \Leftarrow c3\}$
Note that since initial $Source_P$ does not include $p3$ so we cannot use an attack to $c2$ by $p3$.

1. Let $p0$ be a conclusion. Then $Tr = \langle \{p0\}, \emptyset\rangle$.
2. C's next move has two possibilities, that is, to give either $\emptyset$ or $\{\langle c1, p0\rangle\}$.
3. Suppose that C gives $\{\langle c1, p0\rangle\}$. Then, $Tr = \langle \{p0, c1\}, \{\langle c1, p0\rangle\}\rangle$.
4. P's next move has four possibilities, that is, to give either $\emptyset$ or $\{\langle p1, c1\rangle\}$ or $\{\langle p2, c1\rangle\}$ or $\{\langle p1, c1\rangle, \langle p2, c1\rangle\}$.
5. Suppose that P gives $\{\langle p1, c1\rangle, \langle p2, c1\rangle\}$. Then,
$Tr = \langle \{p0, c1, p1, p2\}, \{\langle c1, p0\rangle, \langle p1, c1\rangle, \langle p2, c1\rangle\}\rangle$.
6. C's next move has four possibilities, that is, to give either $\emptyset$ or $\{\langle c2, p1\rangle\}$ or $\{\langle c3, p2\rangle\}$ or $\{\langle c2, p1\rangle, \langle c3, p2\rangle\}$.

[5] This means that even if there are possible counter-arguments, an agent can be silent.

7. Suppose that C gives $\{\langle c2, p1\rangle, \langle c3, p2\rangle\}$. Then, $Tr = \langle\{p0, c1, p1, p2, c2, c3\}, \{\langle c1, p0\rangle, \langle p1, c1\rangle, \langle p2, c1\rangle, \langle c2, p1\rangle, \langle c3, p2\rangle\}\rangle$. Then, since $(p3 \Leftarrow c3) \in Derive_C$, $Source_P$ becomes $\{p0, p1, p2, p3\}$.
8. P's next move has only two possibilities, that is, to give $\{\langle p3, c2\rangle\}$ or $\emptyset$ since $p3$ is now in $Source_P = \{p0, p1, p2, p3\}$ and $\langle p3, c2\rangle$ becomes usable.
9. Suppose that P gives $\{\langle p3, c2\rangle\}$. Then, $Tr = \langle\{p0, c1, p1, p2, c2, c3, p3\}$, $\{\langle c1, p0\rangle, \langle p1, c1\rangle, \langle p2, c1\rangle, \langle c2, p1\rangle, \langle c3, p2\rangle, \langle p3, c2\rangle\}\rangle$.
10. There is no move from both sides so the game is finished.
11. Then, $p3$ is accepted and so $c2$ is not accepted. Then $p1$ is accepted and $c1$ is not accepted. Finally $p0$ is accepted.

In this example, $p3$ is a key to rebut $c2$ and $p3$ was not in initial source but is invoked after $c3$ is made. This invocation is made by a derivation rule $p3 \Leftarrow c3$ (See Fig.1). In the resulting tree, derivation rules play a role of a kind of expansion rules meaning which arguments and attack relations should be added into the initial source of arguments.

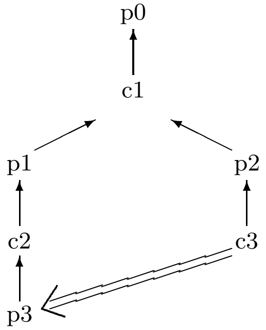


Fig. 1. Representation of Arguments and Derive Relation for Example 1

3 Computing Acceptance in Argumentation Framework

There are many ways to develop an argumentation game tree, but we can show that a final argumentation tree will be unique in any way of developing a tree if both parties eventually give all possible arguments. We call this strategy *eager*, so we can say that an argumentation game tree will converge into one if both agents are eager[6].

From now on, we assume that agents are both eager. Then we can translate an argumentation framework into a logic program in order to compute acceptability of a given argument from the bird's eye view. There is a proposal of computing Dung's argumentation semantics by translating the Dung's

[6] On the other hand, we could define a *lazy* agent which gives only necessary counter-arguments. We could give some analysis about lazy agents as well, but due to limitation of space, we omit the analysis.

framework into a logic program and corresponding answer set of the program with acceptability[Osorio05]. We extend their work by adding an extra condition reasoning about "sources". In order to do so, we introduce new predicate "announced(A)" meaning that an argument A is actually used for building an argumentation game tree. If an argument can be attacked by satisfying the condition that there is an attack relation for the argument and counter-argument is in the source, then counter-argument becomes *announced* to the other agent so that the agent can use other sources of arguments.

Definition 4. *Let* $\langle Arg, Attack, Source, Derive\rangle$ *be an argumentation framework. For* $A \in Arg_P \cup Arg_C$*, we define* $Counter_A = \{B | \langle B, A\rangle \in Attack_P \cup Attack_C\}$*. For each argument* A*, we define the translation of argument* A *to rules of logic programming as follows:*

$$accepted(A) \leftarrow \bigwedge_{B \in Counter_A} not\ (source(B) \wedge accepted(B))^7.$$

Note that if $Counter_A$ *is empty then the above rule becomes* $accepted(A)$*.*
For every $B \in Counter_A{}^8$*,*

$$announced(B) \leftarrow announced(A) \wedge source(B).$$

We also add the following rules for $(A \Leftarrow A_1, ..., A_m) \in Derive_C$*:*

$$source(A) \leftarrow \bigwedge_{i=1}^{m} body_C(A_i).$$

where $body_C(A_i)$ *is defined as follows:*

$$body_C(A_i) = \begin{cases} source(A_i) & if\ A_i \in Arg_C \\ announced(A_i) & if\ A_i \in Arg_P \end{cases}$$

Similarly, we add the following rules for $(A \Leftarrow A_1, ..., A_m) \in Derive_P$*:*

$$source(A) \leftarrow \bigwedge_{i=1}^{m} body_P(A_i).$$

where $body_P(A_i)$ *is defined as follows:*

$$body_P(A_i) = \begin{cases} source(A_i) & if\ A_i \in Arg_P \\ announced(A_i) & if\ A_i \in Arg_C \end{cases}$$

[7] We abuse the notation of logic programming since it contains conjunction of atoms in "negation as failure". However, we can change it into a usual form of logic programming by introducing a rule, $source\_and\_accepted(B) \leftarrow source(B) \wedge accepted(B)$. and the above rule as $accepted(A) \leftarrow \bigwedge_{B \in Counter_A} notsource\_and\_accepted(B)$.

[8] If the parent node is announced and the current node is in the source, then the current node will be announced. This rule expresses the eager strategy of argumentation.

We also add the following for an argument A in the initials source sets:

$$source(A).$$

We also add the following for the conclusion A_0 which is the root of the argumentation game tree:

$$announced(A_0).$$

Note that since there is no loop in the attack set, the above program becomes a locally stratified program so there is a unique minimum model for the translated program[Przymusinska90].

Example 2. Consider the setting of Example 1. The translated logic program becomes as follows:

$$\begin{aligned}
&accepted(c1) \leftarrow \text{ not } (source(p1) \wedge accepted(p1)) \wedge \\
&\qquad\qquad\qquad\quad \text{not } (source(p2) \wedge accepted(p2)). \\
&accepted(c2) \leftarrow \text{ not } (source(p3) \wedge accepted(p3)). \\
&accepted(p0) \leftarrow \text{ not } (source(c1) \wedge accepted(c1)). \\
&accepted(p1) \leftarrow \text{ not } (source(c2) \wedge accepted(c2)). \\
&accepted(p2) \leftarrow \text{ not } (source(c3) \wedge accepted(c3)). \\
&accepted(c3). \\
&accepted(p3). \\
&announced(p1) \leftarrow announced(c1) \wedge source(p1). \\
&announced(p2) \leftarrow announced(c1) \wedge source(p2). \\
&announced(p3) \leftarrow announced(c2) \wedge source(p3). \\
&announced(c1) \leftarrow announced(p0) \wedge source(c1). \\
&announced(c2) \leftarrow announced(p1) \wedge source(c2). \\
&announced(c3) \leftarrow announced(p2) \wedge source(c3). \\
&source(p3) \leftarrow announced(c3). \\
&source(p0). \quad source(p1). \quad source(p2). \\
&source(c1). \quad source(c2). \quad source(c3). \\
&announced(p0).
\end{aligned}$$

Then, we can show that $accepted(p0)$ is derived from the above program.

Theorem 1. *Let $\langle Arg, Attack, Source, Derive\rangle$ be an argumentation framework and A_0 be a conclusion and Tr be a final argumentation game tree w.r.t. the framework for the eager strategy and Pr be a translated logic program from the framework. Then, A_0 is accepted if and only if $Pr \models accepted(A_0)$*

4 Related Works

Several studies have been conducted on argumentation semantics. Dung provided a semantics for a given abstract argumentation framework based on acceptability [Dung95]. He defined several acceptable sets, depending on the range of strength against an attack. Coste-Morquis et al. argued that it is controversial to include both agents' arguments in an extension because this would indicate an indirect attack [Coste-Marquis05]. They defined a new semantics, called "prudent semantics," which does not allow such controversial cases, and compared this with Dung's semantics. Other semantics have also been proposed, such as ideal semantics [Dung06], semi-stable semantics [Caminada06], and others. Baroni et al. compared these types of semantics from the viewpoint of skepticism [Baroni07]. All these semantics involved argumentation systems from a static viewpoint, whereas our proposed semantics is suitable for a dynamic argumentation system.

Cayrol et al. studied how acceptable arguments are changed when a new argument is added to Dung's argumentation system *before an argumentation is executed* [Cayrol10]. Therefore, it is along the line of usual belief revision approach where revision is made before reasoning and revision never occurs during reasoning. In contrast, we focus on addition or arguments during argumentation. So, we believe our approach has more dynamic nature.

García et al. formalized argumentation based on Defeasible Logic Programming (DeLP) [Garcia07]. In DeLP, agent's knowledge base consists of two kinds of rules: strict rules and defeasible rules. The result of argumentation is different depending on which defeasible rules are used. Afterwards, Moguillansky discussed revision of the knowledge base [Moguillansky08]. In his method, after constructing the initial argumentation tree called dialectical tree, knowledge base is changed by extracting defeasible rules and the tree is altered. The goal is to construct undefeated argumentation by selecting suitable defeasible rules. They presented an algorithm for this alteration of the tree and considered a strategy to get the undefeated argumentation. In a series of studies, they formalized several properties in argumentation based on this approach [Lucero09]. Again the revision of knowledge base in their work is made before an argumentation is executed.

Cobo et al. proposed an argumentation framework in which available arguments change depending on time intervals [Cobo10]. In their work, these intervals are given in advance, they did not consider the mechanism by which an argument causes to generate a new argument. In contrast, we focus specifically on the effect of knowledge gained from presented arguments, which is essential in actual argumentation.

Prakken formalized an argument game and showed that counter-argument might not be effective in a game if it is added dynamically and proposed a notion of relevance to make counter-argument effective[Prakken01]. However, in this work, possible arguments are already defined before the game and are never added whereas in our work possible arguments are added according to other party's argument.

Argumentation-based approach is applied to formalize processes appeared in agents communication such as negotiation[Amgoud00]. Considering the effect of the execution of arguments, agents communication are rather related issue, since belief of each agent is updated on receiving information from the other agent. Amgoud proposed the protocol that handles arguments and formalized the case in accepting/rejecting new information [Amgoud00]. She also presented a general framework for argumentation-based negotiation in which agent has a theory and it evolves during a dialogue [Amgoud08]. She considered the knowledge base for each agent separately, as well as its revision by exchanging arguments. The significant difference between her work and ours is that in her approach, an attack relation is increased only between a previous argument and the currently proposed argument whereas in our approach, a dynamic addition of an attack relation does not have such restriction so that we can add any attack relation using *Derive* and *Source*.

5 Conclusion

The contributions of the paper are as follows.

- We give more general framework of argumentation under incomplete information for knowledge integration. We believe that this framework is useful to see how discussions are developed by analyzing how new arguments are introduced.
- We give a computational method of how to decide the acceptability of the arguments using a translation from an argumentation framework to a logic program under the assumption that every possible arguments are made.

As a future research, we would like to pursue the following.

- We would like to introduce the strength of arguments which is related with legal significance.
- We would like to consider how we could apply this framework to reason about a response which could make “a trap” against the opponent where some of opponent responses could cause contradiction in another line of arguments.

References

Amgoud00. Amgoud, L., Parsons, S., Maudet, N.: Arguments, Dialogue, and Negotiation. In: Proc. of ECAI 2000, pp. 338–342 (2000)

Amgoud08. Amgoud, L., Dimopoulos, Y., Moraitis, P.: A General Framework for Argumentation-Based Negotiation. In: Rahwan, I., Parsons, S., Reed, C. (eds.) ArgMAS 2007. LNCS (LNAI), vol. 4946, pp. 1–17. Springer, Heidelberg (2008)

Baroni07. Baroni, P., Giacomin, M.: Comparing Argumentation Semantics with Respect to Skepticism. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 210–221. Springer, Heidelberg (2007)

Bikakis10. Bikakis, A., Antoniou, G.: Defeasible Contextual Reasoning with Arguments in Ambient Intelligence. IEEE Transactions on Knowledge and Data Engineering 22, 1492–1506 (2010)

Cayrol10. Cayrol, C., de St.-Cyr, F.D., Lagasquie-Shiex, M.-C.: Change in Abstract Argumentation Frameworks: Adding an Argument. Journal of Artificial Intelligence Research 38, 49–84 (2010)

Caminada06. Caminada, M.: Semi-stable Semantics. In: Proc. of COMMA 2006, pp. 121–130 (2006)

Cobo10. Cobo, M.L., Martinez, D.C., Simari, G.R.: An Approach to Timed Abstract Argumentation. In: Proc. of NMR 2010, Workshop on Argument, Dialog and Decision (2010)

Coste-Marquis05. Coste-Marquis, S., Devred, C., Marquis, P.: Prudent Semantics for Argumentation Frameworks. In: Proc. of ICTAI 2005, pp. 568–572 (2005)

Dung95. Dung, P.M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-Person Games. Artificial Intelligence 77, 321–357 (1995)

Dung06. Dung, P.M., Mancarella, P., Toni, F.: A Dialectic Procedure for Sceptical, Assumption-based Argumentation. In: Proc. of COMMA 2006, pp. 145–156 (2006)

Garcia07. García, A., Chesnevar, C., Rotstein, N., Simari, G.: An Abstract Presentation of Dialectical Explanations in Defeasible Argumentation. In: Proc. of ArgNMR 2007, pp. 17–32 (2007)

Janjua12. Janjua, N.K., Hussain, F.K., Hussain, O.K.: Semantic Information and Knowledge Integration through Argumentative Reasoning to Support Intelligent Decision Making. Information Systems Frontiers: A Journal of Research and Innovation (2012), doi:10.1007/s10796-012-9365-x

Lucero09. Lucero, M.J.G., Chesñever, C.I., Simari, G.R.: On the Accrual of Arguments in Defeasible Logic Programming. In: Proc. of IJCA 2009, pp. 804–809 (2009)

Modgil09. Modgil, S.: Reasoning about Preferences in Argumentation Frameworks. Artificial Intelligence 173, 901–1040 (2009)

Moguillansky08. Moguillansky, M.O., et al.: Argument Theory Change Applied to Defeasible Logic Programming. In: Proc. of AAAI 2008, pp. 132–137 (2008)

Osorio05. Osorio, M., Zepeda, C., Nieves, J.C., Corte's, U.: Inferring Acceptable Arguments with Answer Set Programming. In: Proc. of ENC 2005, pp. 198–205 (2005), `http://www.lsi.upc.edu/ jcnieves/JCNieves-Publications/Conference/ENC05.pdf`

Okuno09. Okuno, K., Takahashi, K.: Argumentation System with Changes of an Agent's Knowledge Base. In: Proc. of IJCAI 2009, pp. 226–232 (2009)

Przymusinska90. Przymusinska, H., Przymusinski, T.C.: Weakly Stratified Logic Programs. Fundamenta Informaticae 13, 51–65 (1990)

Prakken01. Prakken, H.: Relating Protocols for Dynamic Dispute with Logics for Defeasible Argumentation. Synthese 127, 187–219 (2001)

Rahwan09. Rahwan, I., Simari, G. (eds.): Argumentation in Artificial Intelligence. Springer (2009)

Takahashi11. Takahashi, K., Nambu, Y.: A Semantics for Dynamic Argumentation Frameworks. In: McBurney, P., Parsons, S., Rahwan, I. (eds.) ArgMAS 2011. LNCS, vol. 7543, pp. 66–85. Springer, Heidelberg (2012)

An Open Framework for Exploratory Visual Analysis of Geospatial Data for Winter Road Management

Pavel Moiseets and Yuzuru Tanaka

Hokkaido University, Sapporo, Japan
{moiseets,tanaka}@meme.hokudai.ac.jp

Abstract. With a growing number of vehicles on the roads, traffic problems remain ever present. These problems become especially severe during winter. In this paper, we discuss the first steps for creating a system to manage snow removal in the city of Sapporo. Several questions are raised about the required functionality of such a system and the inherent challenges. We explain how, by designing an explorative framework with the use of the Webble World and a set of custom-made mapping Webbles, we plan to meet these challenges. We discuss how social media can serve as an additional source of information about the city. A way of analyzing the data through dynamic queries is introduced. Finally, to demonstrate the use of our framework, we construct a digital dashboard for data visualization and analysis.

Keywords: Data visualization, Analysis, Webbles, Federation, Social Cyber-Physical System, Winter-Road Management.

1 Introduction

For many years now traffic jams have been a serious problem in big and small cities all around the world. At present there are close to one billion vehicles on the planet. With the world population growing and an expected doubling of the number of vehicles in the following twenty years [1], the importance of problems related to traffic congestion is unlikely to decrease. One such problem is snow and snow removal in winter time. Heavy snowfall hampers not only motor vehicles, but many other modes of transport in the city as well, and that in turn affects the whole city. Just two years ago Federal Government in Washington USA had to shut down for almost a whole week due to buses stopping and snow accumulating on the rail tracks with consequent closure of above ground Metro stations.

Another city with a similar problem, Sapporo in Japan, a metropolis with a population of 1.9 million people, is considered to have some of the heaviest snowfalls among the cities with more than 1.5 million people in the world. With an average snowfall of 6 m, the city government spends more than 150 million dollar per year on snow removal. Such high spending means that there is considerable interest in finding ways to optimize and cut costs. Therefore, as part of general Social Cyber Physical Systems research, we are now conducting studies of snow removal in Sapporo.

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 51–60, 2013.

Social Cyber-Physical Systems (SCPS) feature a close integration and coordination between the system's computational and physical elements in large social systems. SCPS research is a relatively new field, while cyber-physical systems (CPSs) have emerged from previous generation systems, commonly called networked embedded systems. Unlike more traditional embedded systems, a full-fledged CPS is typically designed as a network of interacting elements with physical input and output instead of as standalone devices [2]. Decreasing the costs of snow removal without having a negative impact on the transportation system of the city or, alternatively, improving the traffic situation without spending, leads to one major question: Where and when the snow should be removed? Giving a better solution to this question is the key to success for solving most of snow related traffic problems. And that in turn requires access to various data gathered by smart sensors from all over the city. We believe, managing these sensors in a centralized way and giving meaningful answers based on their data will require the development of a new social cyber-physical system. But that is not the only way such a system could be used. In the future, as the connection between computational and physical elements improves, SCPS has the potential to solve a large variety of social system service problems in a broad array of fields. One other such field is disaster management and response, which is a hot topic in Japan after the 3.11 disaster in 2011. We hope our research may be applied to that field too.

In 2012 MEXT (Japanese Ministry of Education, Culture, Sport, Science and Technology) initiated a new five year national research project on SCPS, an integrated IT platform system for optimizing social system services and for the better sustainability. This project is conducted by a consortium, consisting of NII, Osaka University, Kyushu University, and our group in Hokkaido University. This paper presents our basic architectural idea for the visual analytics of SCPS data with large volume and a variety of different types, especially for the analysis of traffic and meteorological data related to winter road management in Sapporo.

2 Problem Formulation and Methodology

Several types of data were gathered at the start of our research. Among these data are: probe car information, meteorological multi-sensor data from 52 locations in the city, meteorological mesh data, records of snow plowing and removing, complaints and requests from citizens to call centers, and subway passenger records. Coming from different agencies and organizations these data are in a variety of formats and reflect different job conditions and outlooks. Subway data for example, concerning the number of people entering and leaving a station, can be gathered automatically by terminals as people walk by. Call center data on the other hand is inputted manually and is basically little more than text files. Probe cars provide statistical information about average speeds on a number of road segments traveled at any particular timeframe. Their data reflect both huge amount of information handled and the real-time flow from the cars. Such data must be of minimal size while transmitted and stored efficiently without redundancy, but at the same time provide quick results when queried for answers. Computer generated data may be accurate, but hard to read; on the other hand data coming from humans are prone to typing mistakes and omissions. Then again, where sensor or channel of transmission failure is a possibility even computer data may be corrupted. In the end each new source of data brings its own distinct problems. Even when accessed in a retrospective way for preliminary research, a huge amount of work is necessary to aggregate and consolidate them in such a way that different types of data can be combined together. But when development of our integrated system platform is finished, most of the information will be provided in real-time as online services and data sources. To make that possible and to allow for rapid decision making the system must be built from the ground up with requirements for such Improvisational Federation in mind: the system must allow inter-connecting various types of services and data not in a 'planned-for' way, but creatively, with users able to find new use for existing data, not even intended by the developers previously. For that, services and data sources must be wrapped in a unified way to allow interaction. Furthermore, adding new services and data sources is always a possibility, and that is why the system must be dynamic and allow for expansion of its functionality and interface.

While the need to improve snow removal may be clearly stated as a goal, the question of what constitutes an improvement is not so clear. To optimize and improve, some metric of success must be established. But should it be based on the appearance of traffic jams, subway load, number of citizen complaints, or something else entirely? Even if we establish car traffic as a focus, what is an acceptable slowdown for winter time? How should it be measured against the strength of the snowfall? Should some roads or days in the year be considered as critical and given more attention? Snow removal contractors use their own experience and expertise as well as imprecise weather forecasts to make work plans. Snow removal in general is not a well-formed problem, it lacks a single monolithic mathematic formulation of the whole system, however if we focus on appropriate subsystems, we may obtain well-formed problems. In other words, this results in micro analysis instead of macro analysis.

In our search for such appropriate subsystems for mathematical analysis, we need to apply an explorative trial-and-error approach. The search process will consist of repetitions of gathering cyber physical data, applying various analysis and mining tools to them, visualizing the result on a map, planning and making decisions and then giving feedback to the physical world. All these operations should be done improvisationally, since no known established analysis scenario exists. For this approach to work, we need an integrated framework with the following features: 1) Improvisational Federation 2) Rapid map composition 3) Dynamic querying and view modification.

3 Enabling Technologies and Architectural Foundation

We use a previously developed platform, called Webble World, as a base for the system's architecture. This allows us to wrap various tools and data and present them in a uniform way, and enables the Improvisational Federation of these entities.

Webble World deals with Webbles, a type of Meme Media object. Meme Media objects in turn are knowledge media that provide direct manipulation operations for people to reedit and redistribute their content [3]. A Webble may contain more or less anything digitally available. It can, depending on its developer's intentions, carry any kind of digital data, and support any kind of operation and behavior programmable today. It can be stored anywhere on the Internet and then reloaded from most browsers anywhere. Webbles can, regardless of their intended purpose, be customized in many ways by users and combined with other Webbles in order to create compound Webbles or Webble applications, where the primitive building blocks together form a complex tool. Webbles were first introduced by Kuwahara et al. [4].

Each Webble can have any number of slots. A slot is an externally available property, parameter or method controller, through which values may be viewed, exchanged, communicated and modified by Webbles and also by users. Slots can contain virtually any type of data from simple strings and numbers to complex objects, as long as these objects can be serialized in a meaningful way. Some slots are created by default on construction of the Webble; others may be added by users later. To exchange information between two slots of compatible types a channel must established. This channel can be configured to send information in one direction or both.

Webble World itself as a platform can store primitive Webbles or complex application created from them and serves as a "Meme Media marketplace". It allows searching for new Webbles and applications by name, keyword or description and loading them into the browser. Applications, when loaded, can be used as is, modified, or disassembled into components, which in turn can be used to assemble other applications.

Access to the platform and many of the Webbles, developed by us, is available online [5]. Some of the snow removal related data are not open to the public, as we do not have permission to publicize it.

4 GIS Webbles

To visualize spatial data, used in the project, we developed several new Webbles that together can be used to program GIS-like applications. Currently the following Webbles exist: 1) Map 2) Layer 3) Layer List 4) Map Legend 5) Symbol 6) Event

The Map Webble serves as a wrapping for ArcGIS Silverlight API map object. Wrapping an existing API allows us to reuse much of the functionality of that API, so our map already understands many types of layers, drawing functions and so on. Furthermore, since this API is the Silverlight API for ArcGIS, we can also wrap various geoprocessing services from ArcGIS Server later. This Map Webble exposes the map object itself, map's extent, map's list of layers, as well as the canvas that the map belongs to as slots. Although capable on its own due to wrapping an ArcGIS object, the Map Webble is used more like a drawing surface in our architecture, with other Webbles handling most of the functions.

The layers themselves are represented as another class of Webbles. Each Layer Webble exists as a self-contained entity, responsible for its own representation on the map. By adding them as children to a parent Layer List Webble you can populate that list, and those layers will be drawn on the Map Webble, if one is connected to the List.

Many types of layers can be created directly by parsing a special C# XAML notation, inherited from the ArcGIS API. But you can also program your own type of layer with a specialized behaviour. Currently Layer Webbles support several types of open formats for geographical data such as KML, WMS and GeoRSS, as well as Tiled Map layers and Feature layers from ArcGIS Servers. The Webble's slots serve as data sources or parameters for the layer, and there is no clutter between these parameters, as each layer object is separated as a new instance of a Webble. Which slots are available differs depending on the type of the layer. For example WMS and Tiled Map layers require a link to the hosting server, while GeoRSS can be loaded as a string as well. Such a string could be first processed by a Webble specialized in string filtering or XML editing, or even constructed on the fly by a service from the database.

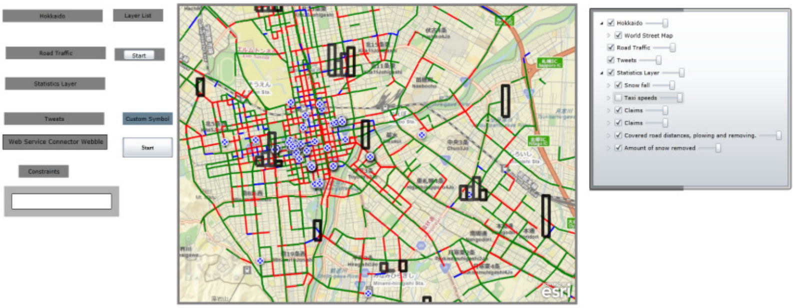


Fig. 1. Map visualization of data from different sources. Data source and map layer Webbles on the left, map legend on the right

A Symbol Webble can be connected to a layer with geographic features such as points, lines and polygons to change the way they are drawn on the map. For example points can be made of different colour and size, changed to pictures, or pie charts with pop-up annotations, embellished with an animation upon mouse over. Lines and polygons can also be customized in the same way, with colour and filling changed.

The Event Webble is attached to the Map Webble to create reactions to specific user actions. For example a user could draw a rectangle on the map by clicking with a mouse, end then get that rectangle's extent from the slot of the Event Webble. That extent can later be fed into another Webble as a parameter for filtering.

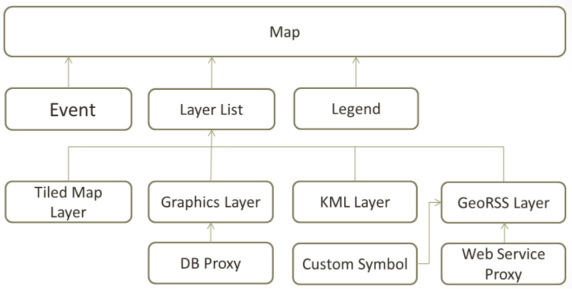


Fig. 2. Sample composition of mapping Webbles

Fig. 1 shows how these Webbles can be used to rapidly (in a matter of minutes) compose a mash-up of different data types (raster tiles, KML, GeoRSS, and custom data with vector features), using various data sources (internal database, ArcGIS server, and Twitter Search API). The map shows traffic data from the probe cars, weather station data, snow removal and call centre complaints data, as well as social media data (from Twitter). Fig. 2 displays the internal architecture of the map part of the same mash-up (each node is a Webble and arrows show the parent-child relations among them). The whole application or its parts can be saved as a composite Webble for later reuse and reediting.

5 Federation with Social Media Services

With the proliferation of social media the amount of useful information that can be gotten from public sources has grown dramatically. Individuals often react faster to events than news organizations or the government. And while it takes time to negotiate for the data from private sources, it takes much less time to obtain data from public sources. With that in mind, for several months, we gathered data from the Twitter Search API with focus on the Sapporo city area (5 kilometer radius around Sapporo station).

There are over 50 thousand tweets made daily in that area, and to analyze them we counted the most frequent words in the tweets. Fig. 3 shows results of this operation for one week in February 2012, using a Chart Webble. The left chart shows the top words in general, while the right chart shows the top words in tweets, mentioning “festival” (the Snow festival was held in Sapporo during that week). What the second chart shows is, that most of the tweets about snow were actually made about the festival, not weather. Also the name of the festival’s mascot “Miku” was mentioned quite a lot.

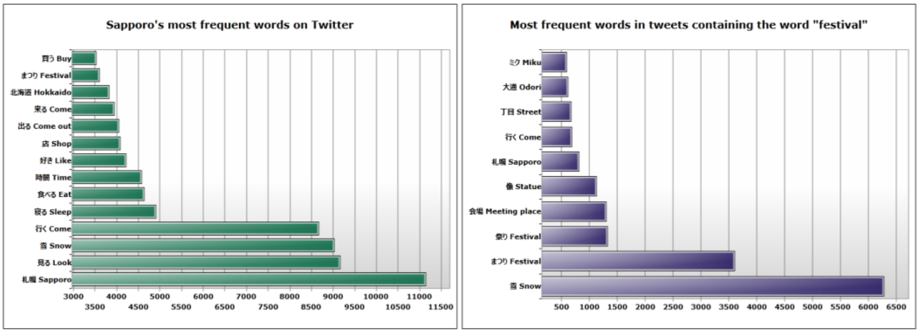


Fig. 3. Popular words on Twitter, February 6 – February 11, 2012

Fig. 4 shows locations of “festival” tweets (on the left) and “Miku” tweets. You can see, that the tweets are concentrated mainly in the Odori Park Street (location of the festival) and specifically the West 5 area (part of Odori Park), where the statue of the mascot was placed, for the “Miku” tweets. During the festival this fell and injured a senior tourist.

It is likely that events important enough to show up on Twitter can affect the road traffic. And as we would like to differentiate traffic jams caused by weather from other types of traffic jams, social media in general can be considered as a useful data source. While we were unable to find any significant amount of tweets about road conditions (such as slipperiness and ongoing road maintenance), the situation can change, if the public knows that the government uses tweets as a source of data. An example of such use of Twitter is the Ushahidi platform [6], which was used in Japan after the Fukushima disaster to gather information from public volunteers.

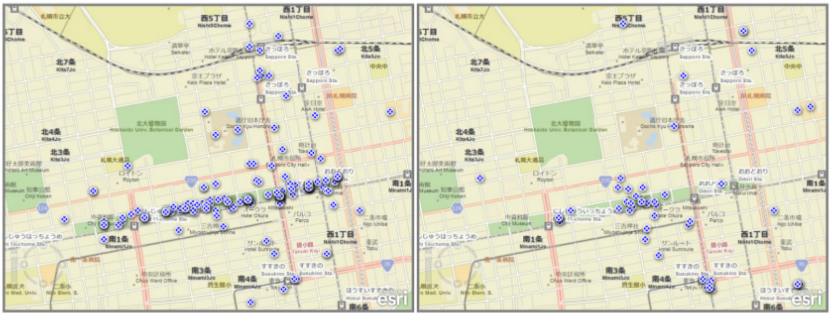


Fig. 4. Tweets about the Snow festival (left) and its mascot (right)

6 Dynamic Queries for Exploratory Visualization

Webble World handles federation of data and analysis tools by wrapping them into Webbles. To allow for fluid application of these tools, where possible, we handle the data in a uniform way as a set of structures, called views. At the moment, views are stored in a relational database as tables, but the same concept could be loosely applied to an object-oriented database or semantic web entities.

Different operations can be applied to the views, resulting in new views. A query operation (query), used on a view (V), requests certain attributes (A) or applies conditions (C) to it. An example of a query would be asking for coordinates and content of all tweets lying in a certain area. Another type of operation are transformations (transform), which change the composition of a view. Sorting the data, or joining different tables are examples of transformations. Analysis operations can also be handled as view based operations. Using Twitter as an example, counting the most frequent words in tweets generates a view with word frequencies, while clustering of roads segments (from a later example), creates a view with clusters of segments. Both of these operations can be thought of as classification (classify) operations, partitioning the data into several classes (tweets containing a certain word or a cluster of road segments). All of the operations mentioned before can be applied sequentially.

$$\text{query}(V_1, A, C) \rightarrow V_2, \text{transform}(V_2) \rightarrow V_3, \text{classify}(V_3) \rightarrow V_4$$

A single view can be visualized (visualize) several times to obtain different types of visualizations (Vis) or to simply look at a different aspect of the same view.

For example, the most popular tweets can be shown on a map or as a chart. The same map can be shown at different levels of zoom. Finally, selection (select) can be applied to a visualization to generate a new condition, with this condition later used to generate a new view. For example, we can select an area of a map, or one of the bars in a chart, to see tweets in that area, or tweets with that word. Once we obtain the new view, we can repeat the analysis operation on it (counting words only in "festival" tweets).

$$\text{visualize}(V_4, \text{chart}) \rightarrow \text{vis}_1, \text{select}(\text{vis}_1) \rightarrow C_1, \text{query}(V_4, C_1) \rightarrow V_5$$

$$\text{visualize}(V_5, \text{map}) \rightarrow \text{vis}_3, \text{classify}(V_5) \rightarrow V_6, \text{visualize}(V_6, \text{chart}) \rightarrow \text{vis}_3$$

In this way the whole process can be represented as a series of succesive views. The current approach is based on evaluation of each query, but in the future it will be extended with caching in order to increase the performance. We think this approach can allow for Exploratory Analysis of the large range of data, used in the project.

7 Digital Dashboard for Exploratory Visual Analytics

We use the same approach, described in chapter 6, in our Open Digital Dashboard System, a Webble World application, currently developed to analyze road traffic changes. Unlike the conventional closed dashboard systems, our system is open for future extension with new tools and data, due to its architecture and use of the Webble World platform.

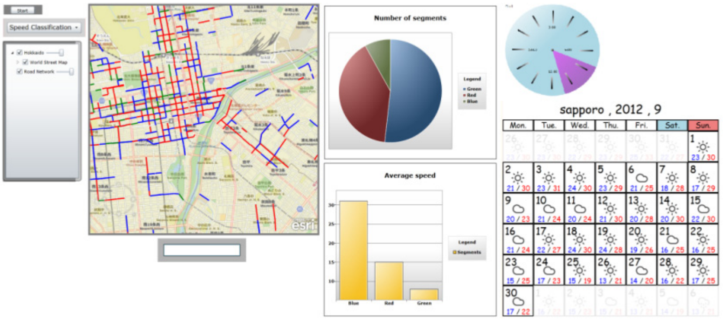
Statistically processed taxi probe car data is provided to us by Fujitsu Co. LTD, with information from about 2000 taxi cars. Traffic data is stored as road segments with known start and end points of the segment, and statistical data about each segment: how many taxis and with what average speed passed through a segment in a 5 minute interval.

Besides simple visualization of the traffic and comparison of winter data against summer data, we use the spherical k-means algorithm [7] (which is a modification of the standard k-means [8]) for clustering the roads segments, based on their average speed values. By clustering the road network into several areas, we hope to identify the problem areas of the network, which snow removal crews would need to focus on.

Fig. 5 shows the current interface of the Dashboard, specifically the part which handles selection and visualization of the data (views). The Calendar Webble shows the general weather information for each day. The Map Webble shows the traffic in the city, while the Chart Webbles (both in a pie chart and a histogram form) are used as auxiliary sources of information. A picker and a text box are used for user input.

The clock, the calendar, and the map (map's area selection event to be more precise) allow using the selection operation, which generate view conditions. For example, we can select the downtown area during the rush hour on a snowy day, or pick a certain road type directly on the map. Analysis and visualization are then performed on the newly selected view. Fig. 6 shows visualization of a series of views, generated by such operations, during a sample data exploration session. Slide 1 shows the first view - a road network, divided according to road types. By selecting one of the types (red) on the map, we receive a condition for that type. By applying that condition to the first view, we generate the second, shown on slide 2 (with only roads of one type). Thereafter we cluster this view by car speeds (the user selects this operation from the picker). The resulting third view is shown on the map (slide 3) and on the chart (slide 4). The chart shows the relative size of each cluster, while the map shows location of segments, colored by cluster type. Again we select one of the clusters on the map, which generates a condition for the fourth view (slide 5). From that condition the histogram (slide 6) is made to show the number of segments which have X cars passing through it (number of cars is the X coordinate, number of segments - Y), by querying new attributes from the fourth view.

The Dashboard is currently used internally as a tool to evaluate the usefulness of various data analysis techniques. The open structure of the Dashboard allows rapidly adding new tools and inventing usage scenarios. Once these scenarios are tested in the Dashboard, they can be further developed into mature applications.



sapporo , 2012 , 9

Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
26 23 / 30	27 23 / 30	28 22 / 29	29 23 / 29	30 25 / 33	31 22 / 27	1 23 / 30
2 21 / 30	3 23 / 31	4 24 / 30	5 23 / 29	6 21 / 25	7 18 / 28	8 17 / 29
9 20 / 23	10 21 / 24	11 20 / 24	12 21 / 30	13 20 / 28	14 20 / 30	15 22 / 30
16 21 / 24	17 22 / 27	18 24 / 30	19 24 / 28	20 19 / 26	21 16 / 25	22 16 / 25
23 15 / 25	24 17 / 23	25 15 / 19	26 13 / 21	27 14 / 20	28 16 / 22	29 17 / 25
30 17 / 22	1 16 / 22	2 15 / 23	3 15 / 21	4 13 / 21	5 14 / 19	6 13 / 21

Fig. 5. Open Digital Dashboard environment for winter road management analysis

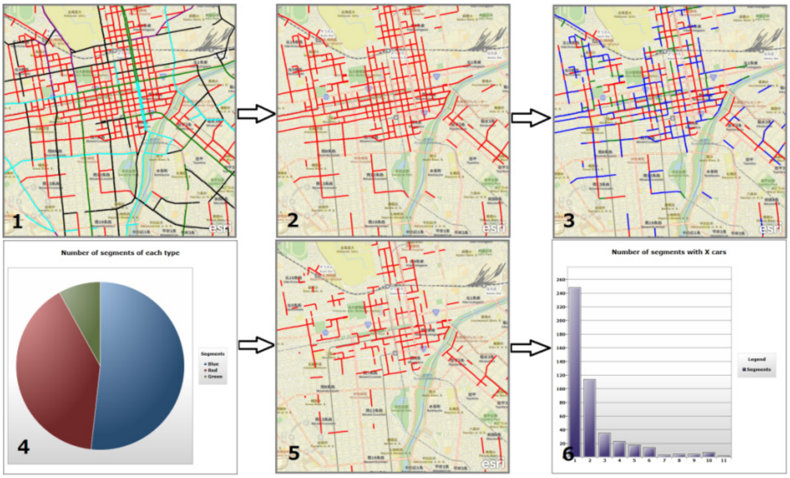


Fig. 6. Elements of the Dashboard at different stages of the analysis

8 Conclusion

Winter road management is a serious task of fundamental importance not only for Sapporo, but for many cities all over the world. Modeling the influence of climate on a complex system is a hard task, with no clear known solution. A survey of Intelligent Trasportation Systems [9] shows that areas such as visual analytics and microblogging have yet to receive enough attention in the field. Past research [10] has already obtained some macro analysis results, showing the influence of temperature and snowfall on traffic in the city, but macro analysis is not sufficient to obtain useful knowledge for the improvement of the traffic situation, that's why we need to expand into micro analysis. We think an explorative and iterative approach is most appropriate for this.

In this paper we laid the ground for building a framework that, we hope, will support the decision making process during road management and snow removal. For this framework we propose using Webble World as a base architecture, with GIS Webbles and Dynamic queries to data, structured as a series of views, for the purposes of analysis and visualization.

We have provided examples of using this framework for building a simple mashup, as well as a more complex application, focusing on Twitter and road traffic data.

In the next step of the project we will utilize our framework to establish meaningful micro analysis scenarios, which will give us useful knowledge for better winter road management.

References

1. Sperling, D., Gordon, D.: Two Billion Cars: Driving Toward Sustainability. Oxford University Press (2009)
2. Lee, E.A.: Cyber Physical Systems: Design Challenges. University of California, Berkeley, Tech. Rep. UCB/EECS-2008-8 (August 2008)
3. Tanaka, Y.: Meme media and meme market architectures: Knowledge media for editing, distributing, and managing intellectual resources. IEEE Press (2003)
4. Kuwahara, M., Tanaka, Y.: Webble world — A Web-based knowledge federation framework for programmable and customizable meme media objects. In: 2010 IET International Conf. on Frontier Computing. Theory, Technologies and Applications, pp. 372–377 (2010)
5. Webble World Portal, `http://cow.meme.hokudai.ac.jp/WebbleWorldPortal/`
6. Ushahidi platform for Japan, `http://www.sinsai.info`
7. Dhillon, J.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Machine Learning 42(1), 143–175 (2001)
8. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A k-means clustering algorithm. Journal of Applied Statistics 28(1), 100–108 (1979)
9. Fei-Yue, W., Kunfeng, W.: Data-Driven Intelligent Transportation Systems: A Survey. IEEE Transactions on Intelligent Transportation Systems 12, 1624–1639 (2011)
10. Munehirom, K., Takahashi, N., Asano, M.: Using Probe-Car Data to Analyze Winter Road Traffic Performance in the Urban Sapporo Area. Monthly Report of Civil Engineering Research Institute, 19–28 (2006)

A Study for Human Centric Cyber Physical System Based Sensing – Toward Safe and Secure Urban Life –

Teruo Higashino and Akira Uchiyama

Graduate School of Information Science and Technology, Osaka University
{higashino,utiyama}@ist.osaka-u.ac.jp

Abstract. Recently, a large number of sensors have been deployed in urban areas, and they can be used for several purposes toward safe and secure urban life. Cost effective sensing and controlling for human activities are becoming possible. As pervasive sensing progresses, Human centric Cyber Physical System (HCPS) will receive much attention where effects of human activities are taken into consideration for designing and developing CPS based social systems. In this paper, we focus on safe and secure urban life and present two case studies: (a) classification of patients in emergency hospitals and (b) estimation of urban pedestrian flows. We also present our testing environment for HCPS.

Keywords: cyber physical system, pervasive sensing, smart city, mobility.

1 Introduction

Recently, a large number of sensors have been deployed in urban areas, and they can be used for several purposes toward safe and secure urban life. Several types of vital sensors have been developed, and they can be used for healthcare and monitoring older people at home. Sensors for environmental monitoring and pedestrian flow sensing have been also designed and developed. Such sensors can be used for pedestrian flow control and energy reduction in urban environments. Using those sensors, cost effective sensing and controlling for human activities are becoming possible. As pervasive sensing progresses, Human centric Cyber Physical System (HCPS) will receive much attention where effects of human activities are taken into consideration for designing and developing CPS based social systems.

In this paper, we focus on safe and secure urban life, and present two case studies: (a) classification of patients in emergency hospitals and (b) estimation of urban pedestrian flows. In the first case study (a), we have developed small-sized electronic triage tags and collected vital signs of critical patients at waiting rooms in emergency hospitals for more than two years [1]. We have stored monitored vital signs with their ages, chief complaints, past records and physical conditions, and decided thresholds for issuing warning representing that the necessity of emergency care for designated patients becomes rather high. Warned

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 61–70, 2013.

patients are given high priority for emergency care. Our experiments show that the ratio of sudden change of physical conditions of patients at the waiting room of a hospital can be reduced to about one third. In the second case study (b), we have developed human mobility sensing methods in urban areas [2,3]. Our method derives expected flows of pedestrians from observations/sensing of average pedestrian densities.

Finally, we have developed a simulator called HumanS for evaluating how human activities affect pedestrian flows in urban areas and energy consumption of buildings, guidance of pedestrians in shopping malls and evacuation planning in underground cities [4]. Location information about people can be utilized for pedestrian navigation in different situations. There are many types of sensors such as laser-range scanners, infrared-based position sensitive detectors and image/thermo analyzers and so on. Sensor deployment affects accuracy of crowd sensing. Here, we explain how HumanS can evaluate the performance of crowd sensing for a given sensor deployment. We also introduce the possibility of evacuation planning in underground cities. In addition, we present our testing environment for reproducing rescue operations in urban areas on cyber space.

2 Related Work

In US, Cyber Physical System (CPS) based researches funded by NSF are actively studied where new types of IT technologies necessary for energy reduction, future transportation and healthcare are developed. Recently, as smartphones become popular, sensing and controlling for human activities using smartphones are becoming possible. Using such functions, Human centric Cyber Physical System (HCPS) are actively studied where pervasive sensing is one of key technologies. Several research works concerning with pervasive sensing have been done. For example, in CenceMe project [5] smartphones are used for detecting user activities. In CitySense project [6], crowd-sourced sensing information is used for analyzing human behavior and social activities in urban areas. In [7], a people-centric sensing is proposed where sensors on smartphones work together with fixed sensors in the environments. It also discusses the sharing of data among friends and communities, as well as opportunistic sharing of the sensing tasks. In [3], a positioning system for people-centric navigation using smartphones is proposed where relative positions among surrounding people are discussed.

When we consider human centric CPS, it is important how we can precisely estimate human behavior and social activities. Especially, if we develop smartphone-based MANET applications used in urban areas, accurate mobility generation of humans becomes a key technique. It is well known that node mobility and density affect the performance of MANET applications [8,9], and many mobility models have been proposed so far [10]. Random-based mobility models such as Random WayPoint (RWP) model and Random Direction (RD) model are very popular, and many analytical researches have revealed their properties (e.g. see [11]). On the other hand, if we want to design MANET applications for pedestrians with smartphones and/or running vehicles in urban areas, we need

Table 1. Categories and Severity

Category	Severity
Black (O)	dead or dying: no care
Red (I)	life-threatening: need immediate care
Yellow (II)	treatment and transportation can be delayed
Green (III)	those with minor injuries: less urgency

more realistic mobility. In [2], we have shown that there are large variations for performance and packet loss rates of wireless communications depending on node density distributions. In Delay Tolerant Networks (DTNs), it is also known that node mobility and density strongly affect the reliability and performance of DTN applications (e.g. see [9]).

Here, we focus on human centric CPSs in urban areas, and introduce how such human centric CPSs are used for safe and secure urban life.

3 Advanced Electronic Triage System

Triage is a process of prioritizing patients based on the severity of their physical condition when many persons are simultaneously injured in disaster. This process is used for rationing patient treatment efficiently when medical resources are insufficient for all to be treated immediately. The categorization of patients based on the severity of their injuries can be aided with the use of printed triage tags or colored flagging. In Japan, triage was applied for the first time in April 25, 2005 for a big derailment accident where totally 107 persons were died and 562 passengers were injured. At that time, triage was done based on START (Simple Triage And Rapid Treatment) protocol. It categorizes each patient into one of four categories shown in Table 1. A four-color paper tag is attached to each patient to indicate the determined category.

Simple triage is usually used when a mass-casualty incident occurs. It sorts injured persons into those who need critical attention and immediate transport to the hospital and those with less serious injuries. Even if we can sort injured persons using the printed triage tags, it is not easy for rescue parties and doctors to precisely find where patients with serious injuries are lying in the disaster area.

Thus, we have designed and developed an electronic triage system called eTriage system (see Figure 1) [1,12]. The proposed eTriage system senses physical conditions of those injured persons using electronic triage tags and collects their sensed data in ad-hoc wireless communication. The electronic triage system presents dynamic change of injured persons' location and physical conditions on monitors in real time. Our research aims to improve efficiency of rescue operations in disaster using the proposed advanced wireless communication technology.

The electronic triage tag automatically monitors each patient's SpO_2 (blood oxygen level) and pulse rate by inserting a finger of the patient into the electronic triage tag. The monitored vital signs are periodically transmitted to the server via a ZigBee/WiFi based ad-hoc wireless network constructed by electronic triage tags and pocket routers temporally installed in road cones at the

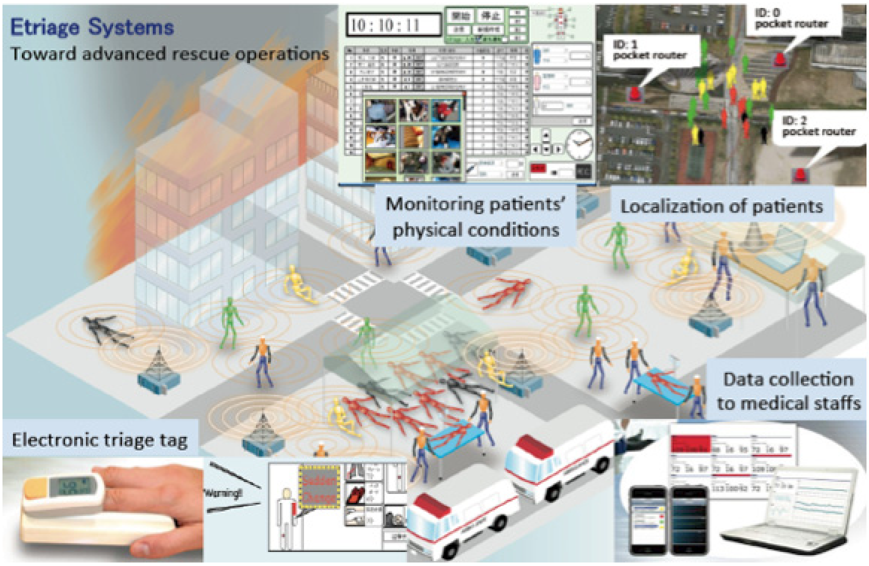


Fig. 1. eTriage system

target area. The gathered vital signs are recorded as database at a server in the disaster area and can be accessed via Web browsers from all the PCs and mobile devices in the disaster area. The system can alert medical staffs to dangerous conditions of patients based on the collected vital signs if the condition of a patient becomes worse suddenly. The doctors and nurses bring mobile devices such as iPod touch (or iPad). They put iPod touch on their arms. The vital signs of patients are shown as line graphs on the display of iPod touch. Each target patient is recognized using human body communication. If a doctor touches any place of the body of a patient, the ID of his/her electronic triage tag is automatically informed to the doctor's iPod touch.

Medical staffs in emergency and critical care centers of many hospitals have difficulty in dealing with a great number of patients especially in holidays and after hours. The proposed eTriage system can be expected to use not only for disaster situations but also for emergency and critical care centers in hospitals. To see the efficiency, we conducted test operation at the emergency and critical care center in Juntendo University Urayasu Hospital for two years from January 2009 to December 2010. In the operation, an electronic triage tag was attached to a patient whose symptom could empirically lead to sudden change to worse conditions, and the vital signs of such patients were monitored in real-time.

For patients visiting the emergency and critical care center by themselves in year 2008, we recorded patients' claims, vital signs, their medical history and clinical diagnosis when the patients entered the hospital. Especially, for cases that the vital signs of patients were suddenly changed during they waited at the emergency and critical care center, we have analyzed their situations in detail. It is known that about 4.9 million people in Japan visited emergency hospitals in 2006. The number increases 50 percent than 10 years ago. However, it is also known that most of their conditions are mild. The conditions of very few patients (e.g. 1 or 2 percent) are critical. In the case of Juntendo University Urayasu Hospital, only 1 percent of patients who come to the hospital by themselves have entered to its emergency ward. The rest of patients return to their home after treatment or enter general wards. On the other hand, the physical conditions of 0.05 percent of patients are suddenly changed during they wait for consultation at the emergency center. The average age of those patients is about 60, and the main diagnoses are heart trouble, brain blood vessel trouble and digestive organ trouble. The physical conditions of old aged persons whose main claims are "nausea" and "dizziness" might be suddenly changed. From such observation, from January 2009 at the emergency and critical care center of the hospital, our eTriage system has been used for monitoring patients whose ages are more than 50, whose main claims are breast ache, difficulty in breathing, conscious disorder, nausea, dizziness, headache and stomachache, and so on. When our eTriage system detects that the physical condition of any patient is changed based on the collected vital signs, it alerts medical staffs to dangerous conditions of the patient. From two years' experiments, the ratio of sudden change of physical conditions of patients at the waiting room is decreased to about one third. We believe that this ratio indicates the effectiveness of monitoring vital signs at emergency and critical care centers in hospitals as well as disaster areas although there are some other factors such as improvement in management of patients.

4 Urban Pedestrian Flow Mobility

In recent years, sensors have been connected via wired or wireless networks to the Internet infrastructure and the collected data are analyzed to capture the features of the physical world. In particular, sensing in urban environments is significant. For example, sensing and detecting activity of people in public spaces such as shopping malls and event exhibitions by laser-range scanners, image sensors and so on will be helpful to understand their interests. The location of people can be utilized for pedestrian navigation in different situations.

In our previous work in [2], we have proposed a method to create Urban Pedestrian Flow (UPF) mobility. The method reproduces walking behavior of pedestrians in city sections, shopping malls and so on. Given the average densities of pedestrians on certain streets, which can be easily obtained by fixed point observations/sensing, and a set of possible walking paths which pedestrians are likely to follow, the method determines expected flows of pedestrians using linear programming (LP) techniques where the maximum error between the observed density and the derived density is minimized so that we can reproduce realistic movement of pedestrians. We have measured the average densities of pedestrians on 33 streets in a 500 m * 500 m area in front of a large train station in Osaka for about a half hour. The maximum error between the observed densities and the derived densities was only 9.09 percent. In Figure 2, we show three mobility models: (a) Random WayPoint (RWP) mobility, (b) Random WayPoint with

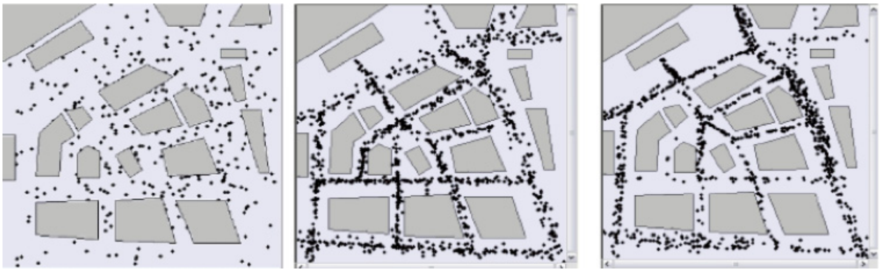


Fig. 2. Mobility in urban areas

obstacles (RWP/ob) mobility, and (c) Urban Pedestrian Flow (UPF) mobility. The node densities in RWP mobility and RWP/ob mobility are almost uniform. On the other hand, UPF mobility reflects realistic node density distribution and variation in urban areas. We have also developed a mobile wireless network simulator called MobiREAL [2] for simulating MANET applications with UPF mobility scenarios.

5 Testing Environment for HCPS

We have developed a simulator called HumanS for evaluating how human activities affect energy consumption of buildings, guidance of pedestrians in shopping malls and evacuation planning in underground cities [4]. Lights and air conditions in buildings can be managed by using more efficient energy management based on detailed information about people's indoor location. HumanS is a multiagent simulator that works with geographic information system (GIS), and has the following features.

(1) It models realistic behavior of pedestrians (human agents), which has a significant impact on the performance of such sensing systems. It determines moving paths of human agents from given (or observed) origin-destination pairs.
(2) It provides several sensor models that can be placed on GIS maps. There are many types of sensors such as laser-range scanners, infrared-based position sensitive detectors and image/thermo analyzers to detect objects. We provide generalized models of those sensors, and such sensor entities can be placed on any location in given maps to detect the presence of people in the sensing area.
(3) It stores and manages scanning data in a GIS database by appropriately tagging their time and location. Spatial and temporal queries can be processed using the GIS database.
(4) It visualizes sensor location, sensing regions and human location and mobility on GIS map (see Figure 3). It helps intuitive awareness, recognition and analysis of events in simulation, such as miscount of people in some crowded regions.

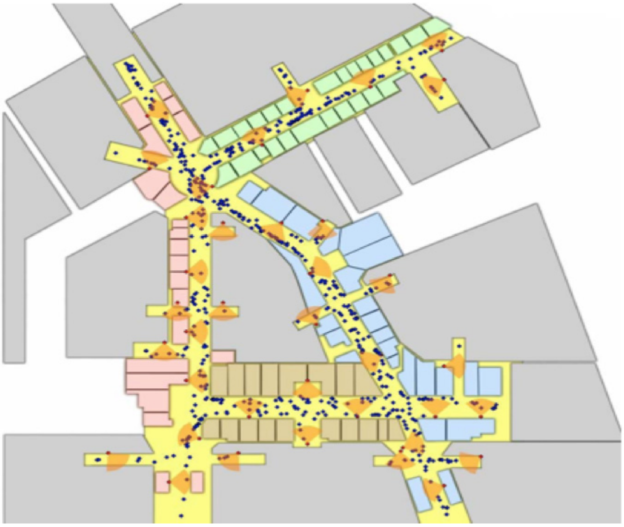
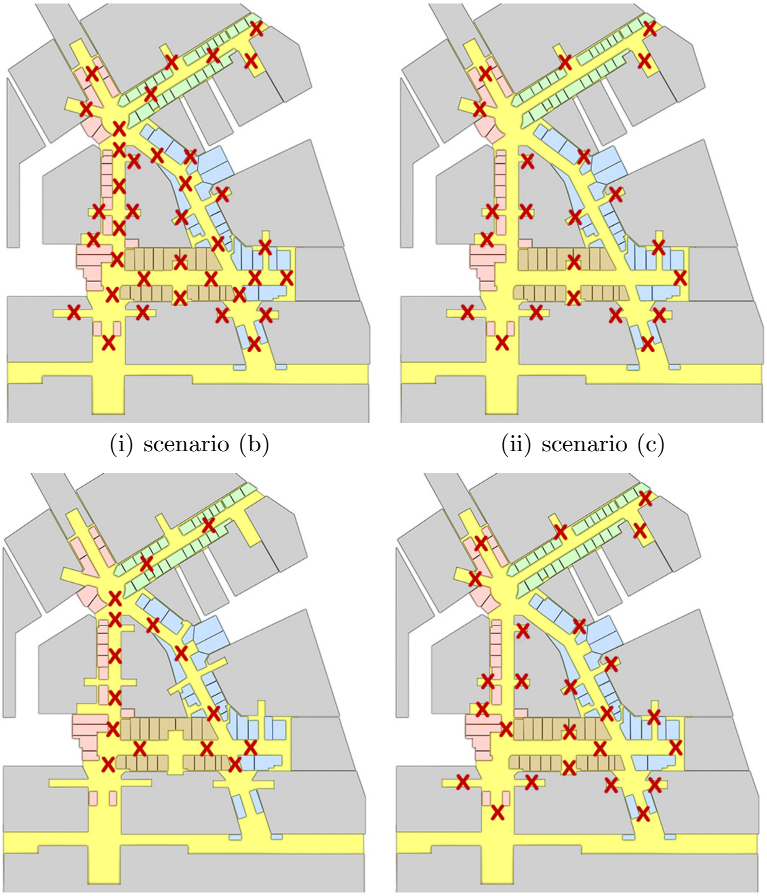


Fig. 3. Screenshot of HumanS simulator

We have evaluated the capability of HumanS by a case study. We have generated a realistic mobility of human agents in the underground city of Osaka downtown [13] based on the real population data [14]. Then using our sensor model of laser-range scanners, the pedestrian flow estimation method discussed in the previous section [2] has been simulated to test its performance. We have verified that different types, placement and number of sensors resulted in different results, and have confirmed that such assessment can be done in a more efficient way than field experiments.

We have used laser range scanners (LRS) with 10m maximum range, 60 degrees of angle (with three degree angle resolution) and two seconds scan interval. Since the LRS model cannot detect objects behind others, there always exists miscounting of people. To observe the impact of sensor deployments, we have considered several sensor deployments for the underground city in Figure 3 and carried out simulations. In the underground city, there are 37 pathways. We have compared the following five scenarios (see Figure 4).

(a) 37 “perfect” sensors are deployed on all pathways. This means that every pathway has a sensor and it can precisely count the number of people in the sensing region without errors
(b) 37 LRSs are deployed on all pathways (every pathway has one LRS)
(c) 22 LRSs are deployed on pathways that are connected to entrances (only pathways that are close to origin/destinations are monitored by LRSs)
(d) 15 LRSs are deployed on pathways that are not connected to entrances (only pathways that are far from origin/destinations are monitored by LRSs)
(e) Additionally three LRSs are deployed to scenario (c) (the three LRSs were manually deployed and they are deployed near branch points of pathways)



(i) scenario (b)

(ii) scenario (c)

(iii) scenario (d)

(iv) scenario (e)

Fig. 4. Sensor Placement

Scenario (a) has achieved the best accuracy of the estimated density. The average error of the estimated density is about 10 percent of the real density while that of scenario (b) is about 17 percent. It is clear that this 7 percent difference is due to difference of sensor capability, which is significant in sensing systems. Actually only 11031 out of 14013 people have been detected by LRSs in scenario (b) (i.e. the detection ratio is about 79 percent). In scenario (c), we can see larger errors on pathways without LRSs. However, the average error of the estimated density is about 28 percent. On the other hand, that of scenario (d) is about 132 percent although more people could be detected than scenario (c). Finally, according to the simulation result of scenario (c), we have tried to improve the accuracy by manually adding a limited number of LRSs in scenario (e). The branch points of pathways give large influence for estimating the density of pedestrians. Based on this observation, we add only three LRSs. Then, we could substantially improve the accuracy of the estimated density where the average error is reduced to 14 percent. This error ratio is close to the performance in scenario (b). This is a surprising finding and contributes to more cost-efficient monitoring of pedestrians activity.

Finally, we introduce the support functions for evacuation planning in underground cities such as Figure 3. In order to make an evacuation plan for an underground city and improve the performance of its rescue operations, it is important to repeat training in real environments based on the evacuation plan. However, it takes too much cost to carry out such training in real environments. So, we have developed a testing environment for rescue support applications on cyber space [15]. In the testing environment, real terminals can be connected to virtual environments based on a given GIS model and pedestrian behavior model. Our simulator captures packets sent from real terminals and passes them into the real-time simulator. Also, when packets are sent from nodes in the simulation, they are translated into real packets and sent to the corresponding real terminals. By using such a testing environment, we can test the rescue operations with real terminals as if they are connected via a real ad-hoc network constructed in a disaster area. Originally, the virtual space has been constructed in SecondLife [16] provided by Linden Research, Inc., which has rich interfaces for controlling virtual objects from outside. Now, we are planning to construct the virtual space on a more realistic 3D space using AR techniques.

6 Conclusion

In this paper, we focus on Human centric Cyber Physical System (HCPS) and represent two case studies using pervasive sensing techniques. We have shown how technologies for sensing human activities and vital signs can be used for safe and secure urban life. As our future work, we are planning to develop more realistic virtual space using AR techniques so that we can discuss about performance and reliability of several types of HCPSs which can be used as systems for energy reduction, future transportation and healthcare more precisely.

References

1. Higashino, T., Uchiyama, A., Yasumoto, K.: eTriage: A Wireless Communication Service Platform for Advanced Rescue Operations. In: Proc. of ACM Workshop on Internet of Things and Service Platforms (IoTSP 2011) (2012) (Keynote Speech/Invited Paper)
2. Maeda, K., Uchiyama, A., Umedu, T., Yamaguchi, H., Yasumoto, K., Higashino, T.: Urban Pedestrian Mobility for Mobile Wireless Network Simulation. Ad Hoc Networks 7(1), 153–170 (2009)
3. Higuchi, T., Yamaguchi, H., Higashino, T.: Clearing a Crowd: Context-supported Neighbor Positioning for People-centric Navigation. In: Proc. of 10th IEEE Int. Conf. on Pervasive Computing, pp. 325–342 (2012)
4. Kanaya, T., Hiromori, A., Yamaguchi, H., Higashino, T.: HumanS: A Human Mobility Sensing Simulator. In: Proc. of 5th IFIP Int. Conf. on New Technologies, Mobility and Security (2012)

5. Miluzzo, E., Lane, N.D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S.B., Zheng, X., Campbell, A.T.: Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In: Proc. of 6th ACM Conference on Embedded Networked Sensor Systems (SenSys 2008), pp. 337–350 (2008)
6. Murty, R., Mainland, G., Rose, I., Chowdhury, A.R., Gosain, A., Bers, J., Welsh, M.: CitySense: An Urban-scale Wireless Sensor Network and Testbed. In: Proc. of 2008 IEEE Int. Conf. on Technologies for Homeland Security (2008)
7. Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musolesi, M., Fodor, K., Ahn, G.-S.: The Rise of People-Centric Sensing. IEEE Internet Computing 12(4), 12–21 (2008)
8. Royer, E.M., Melliar-Smith, P.M., Moser, L.E.: An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks. In: Proc. of IEEE Int. Conf. on Communications (ICC 2001), pp. 857–861 (2001)
9. Zhang, X., Kurose, J., Levine, B.N., Towsley, D., Zhang, H.: Study of a Bus-based Disruption-Tolerant Network: Mobility Modeling and Impact on Routing. In: Proc. of ACM Int. Conf. on Mobile Computing and Networking (MobiCom 2007), pp. 195–206 (2007)
10. Tracy, T., Jeff, B., Vanessa, D.: A Survey of Mobility Models for Ad Hoc Network Research. Wireless Comm. & Mobile Computing (WCMC) 2(5), 483–502 (2002)
11. Chu, T., Nikolaidis, I.: Node Density and Connectivity Properties of the Random Waypoint Model. Computer Communications 27(10), 914–922 (2004)
12. eTriage System HP: http://www-higashi.ist.osaka-u.ac.jp/research/e-triage.html
13. Diamor Osaka: http://www.diamor.jp/lang/en/
14. Population Census in National Survey of Japan: The Statistics Bureau and the Director-General for Policy Planning of Japan, Tech. Rep. (2005) (in Japanese), http://www.stat.go.jp/data/kokusei/2005/jutsu1/00/03.html
15. Nakata, K., Maeda, K., Umedu, T., Hiromori, A., Yamaguchi, H., Higashino, T.: Modeling and Evaluation of Rescue Operations using Mobile Communication Devices. In: Proc. of 23rd ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PADS 2009), pp. 64–71 (2009)
16. Second Life: http://secondlife.com/

DiseaseMedia: An Information System for Helping Diagnosing and Treating Rice Diseases

Richard Chbeir[1], Asanee Kawtrakul[2],
Dominique Laurent[3], and Nicolas Spyratos[4]

[1] LIUPPA-Université de Pau et des Pays de l'Adour, Biarritz, France
[2] NaiST-Kasetsart University, Bangkok, Thailand
[3] ETIS-CNRS-ENSEA-Université de Cergy-Pontoise, Cergy-Pontoise, France
[4] LRI-CNRS-Université Paris Sud, Orsay, France

Abstract. We present an information system supporting the diagnosis and treatment of rice diseases. The heart of the system is a software mediator allowing farmers to formulate distant queries regarding a disease and responding to farmer queries by recommending a treatment of the disease if one exists. The disease diagnosis and the recommended treatment are based on expert knowledge stored in the mediator database. The processing of farmer queries may involve direct access to the mediator by farmers, online communication between the mediator and the experts, or direct dialog between the farmer and an expert. Our information model is generic in the sense that it can be used to support the needs of farmers in a variety of similar environments, with minor changes. The system presented in this paper is currently under development as part of a Franco-Thai project[1] and aims to assist farmers in the quick diagnosis of rice diseases and their treatment.

1 Introduction

The objective of this paper is to propose a generic, computer aided system supporting the communication between farmers and experts of rice diseases. The case study that motivates our work is that of rice diseases in Thailand, where the government deploys a substantial effort to help farmers cope with such diseases. Rice is the main agricultural product of Thailand and rice farmers represent 66% of the 5.7 millions of agricultural households. However, there are several key issues affecting agricultural rice production such as including: (1) disaster from the pest both insects and diseases that cause the damages of production, (2) the climate change that causes both drought and/or flood affecting lower yield, (3) the adjustment of resources (such as fertilizer) and wages affecting the higher expense, etc. Moreover, the average age of the farmers is being currently 58 years old which will certainly cause a generation gap in the farmer population in the near future. Regarding rice diseases, the main problem is that there is currently no systematic monitoring and diagnosis of rice diseases in Thailand, and no

[1] Supported by the French-Thai EGIDE project 25659SB.

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 71–80, 2013.

organized communication between farmers and experts in order to provide timely treatment to diagnosed diseases. This situation can lead to: (*i*) catastrophic rice seasons (if a disease is not diagnosed early enough during the evolution of rice plants or if a diagnosed disease receives an ad hoc treatment), and (*ii*) money wasting (because of the cost of treatments).

To cope with these difficulties, efforts are made by the Thai and local governments to help farmers identify diseases ([3]). One significant effort in this respect consists in promoting the creation of Coops and selecting and training (at least) one farmer in each Coop. Such a trained farmer is called a Cyberbrain ([4]) and his role is to be able to identify as many diseases as possible and to choose appropriate treatments. However, despite these efforts, the use of technology at the right time with the right knowledge for the right situation is still a goal to be reached, in order to reduce costs and improve productivity, while taking into account sustainable agricultural development. Setting up a software system whereby farmers can easily and remotely formulate queries to experts of plant diseases, and having those experts advise the farmers on the appropriate treatment, is of paramount importance as it can save the crops of a whole year. The objectives of our study are as follows:

- Easing the formulation of queries by the farmer query and providing quick responses (either an appropriate treatment or interaction with an expert);
- Helping detect relevant treatments;
- Tracing rice disease evolution across the country as well as the relevance and the consequences of recommended treatments.

Roughly speaking, the information system that we propose will allow a rice farmer or a cooperative of rice farmers (hereafter called Coop) to easily issue a query to an expert of rice diseases, or a pool of experts through a software mediator as shown in Figure 1. What we call a farmer query consists of one or more pictures taken by the farmer together with a checklist of symptoms. The pictures are in digital form and so is the checklist which the farmer can complete with a few clicks (possibly with assistance from his Coop).

A typical session with our system consists of a farmer issuing (mainly with his smartphone) a query to the mediator, and the mediator processing the query as follows: if the answer can be obtained from the mediator database then it is sent to the farmer, otherwise the mediator performs the following tasks:

1. selects an expert and passes to him the farmer query,
2. initiates a direct interaction between the farmer and the expert, and
3. provides support to this interaction (in a way that we shall see shortly).

A typical scenario in the context of our system can be described as follows: Observing the overall color of a field, a farmer suspects that some disease is affecting the rice plants. The farmer then takes one or more pictures of diseased leaves using a cellular phone or some specialized photographic equipment. Using these pictures, the farmer accesses the mediator and issues a query by visual example(s) to which the mediator responds by returning the list of similar pictures stored in the mediator database. Following the mediators response:

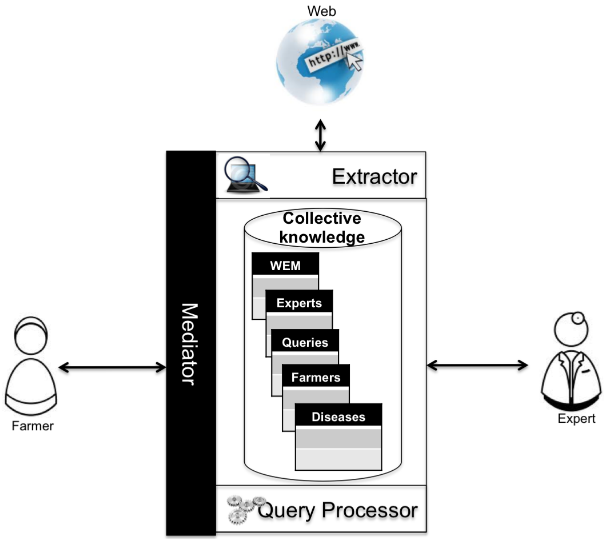


Fig. 1. An overall view of the system

- If the farmer identifies a matching picture in the list, then the farmer clicks on it and the mediator returns the appropriate treatment for related disease (without intervention of an expert).
- If the farmer identifies several possible pictures, the mediator sends back to the farmer a list of symptoms characterizing the diseases related to the selected pictures.
  - if the farmer fills in and returns a list of symptoms that are sufficient to identify a single disease from the mediator database then the mediator returns to the farmer the appropriate treatment for that disease (without intervention of an expert)
  - else the mediator selects an expert, transmits to him the query, and initiates a direct dialogue between the farmer and the expert in an effort to have the disease identified.
- If the farmer does not select any pictures provided by the mediator, then this latter identifies an expert from its database, transmits to him the query, and initiates a direct dialogue between the farmer and the expert in an effort to have the disease identified.

The choice of an expert by the mediator is based on the information contained in the farmer's query, as well as on information stored in the database. In other words, the mediator plays the role of the expert whenever possible, and passes responsibility to an actual human expert, otherwise. The idea is to call on the expert as infrequently as possible, in order to save time and to reduce cost.

We emphasize that the data provided by the farmer to the mediator can be obtained with minimal effort: the checklist can be filled with a few clicks and pictures of an affected plant can be taken easily using a cellular phone or a CyberScan (a camera-like tool specifically developed for the project). Both data can be sent to the mediator using the cellular phone, and no other data is required on the part of the farmer.

The rest of the paper is organized as follows. In Section 2, we present the mediator; in Section 3, we describe the information flow among the main actors of the system; and in Section 4, we offer concluding remarks and suggestions for future work.

2 The Mediator

The heart of the mediator is a database where all the required data is stored; this data come from three sources: the farmers, the experts and a Web Extraction Module (called "Extractor" in Figure 1). Apart from query processing and database maintenance during updates, the mediator provides services which do not concern individual queries or updates but are related to general knowledge regarding rice diseases. In what follows, we present in more detail the mediator database, the Web Extraction Module and the services that the mediator offers to its users, assuming the following:

– *Visual distance function:* In order to compare two pictures, we assume that the mediator is equipped with a distance function [1] able to compare visual features (e.g., colors) described via one or many descriptors (such as color distribution, histograms, dominant color(s), etc.)
– *Semantic-based distance function:* In order to compare two sets of symptom indicators, the mediator is able to semantically compare them using a given rice-disease ontology (e.g., Agrovoc [2]).
– *Farmer knowledge and equipment:* We assume that farmers have smart phones (able to capture pictures, use 3G network, etc.) and are able to provide (alone or with the help of Coop) the indicators requested by the expert.

2.1 The Mediator Database

The mediator database stores data coming from the two authorized actors accessing the mediator, namely the farmers and the experts, as well as from a specialized module (the Web Extraction Module) that crawls the web in search of new rice diseases and their treatment. In our project, we have chosen a relational database to store the data. The database schema consists of the following five tables (the keys of each table are underlined):

```
 QUERIES (QID, ChecklistURI, PicturesURI, QDate, FarmerID,
         QStatus, FeedbackURI)
DISEASES (DiseaseID, PicturesURI, ChecklistURI, TreatmentURI)
 FARMERS (FarmerID, CyberbrainID, FarmerURI, CyberbrainURI)
 EXPERTS (ExpertURI, DiseaseID)
     WEM (ExtractionID, DiseaseURI, Status)
```

We note that the key of the EXPERTS table is the pair ⟨ExpertURI, DiseaseID⟩ comprising both attributes of this table, meaning that a person can be expert for more than one disease, and conversely, there might be more than one expert for a given disease.

The QUERIES table stores all farmer queries sent to the mediator together with some auxiliary information. When a farmer submits a query to the mediator, a tuple is inserted in the QUERIES table comprising the following values:

- QID: A system allocated identifier for the submitted query; the value of this attribute cannot be NIL;
- ChecklistURI: A set of attributes which represents the main classes of interest to the expert (such as Region, Province, Soil Type, Weather information, Varieties, Stage of rice growth, previous diseases, etc.). Their values are those indicated by the farmer (with the help of the Coop) while completing the check list of symptoms. We note that these values are standardized and are presented to the farmer in the form of a pop-up menu in which the farmer can click one of the values (or NIL);
- PicturesURI: A system allocated URI pointing to a page containing the pictures sent by the farmer; the value of this attribute is NIL, if no pictures are included in the query. When a direct dialogue between the farmer and the expert is necessary, the pictures contained in PicturesURI assist the expert in identifying a disease;
- QDate: The date of issuing the query; the value of this attribute cannot be NIL;
- FarmerURI: A system allocated URI pointing to a page containing contact information of the farmer who submitted the query (such as name, mailing address, email, and phone); the value of this attribute cannot be NIL;
- QStatus: The status of the farmer query which can be one (and only one) of the following keywords:
  - not-yet-processed, if the processing of the query has not started yet
  - unknown-disease, if the expert cannot identify the disease
  - answered, if the mediator sent a treatment to the farmer
  - feedback pending, if the mediator sent the feedback request but has not yet received the farmers answer
  - closed, if the mediator has received and stored the feedback of the farmer;
- FeedbackURI: The feedback of the farmer regarding a recommended treatment (it can be in the form of free text).

Note that the non NIL values of the checklist are used by the mediator in order to filter the DISEASES table, and the result of this filtering determines whether there is zero, one or more diseases that correspond to the checklist(s) (zero, one or more tuples in the result respectively). In case the result contains a single tuple, then the DiseaseID of this tuple serves to determine the corresponding treatment from the DISEASES table.

The DISEASES table stores all disease descriptions and their corresponding treatments, as validated by the experts. These inputs are obtained in one of three ways as the result of (*a*) a farmer/expert interaction, (*b*) newly acquired knowledge by the experts, and (*c*) from the Web Extraction Module after quality control by an expert. A tuple inserted in the DISEASES table consists of the following attribute values:

- DiseaseID: A system allocated identifier for the disease;
- PicturesURI: A system allocated URI pointing to the page containing one or more pictures corresponding to the disease; these pictures can be either the ones taken by the farmer or those coming from the WEM table (in either case, after validation by the expert);
- ChecklistURI: A system allocated URI pointing to a page containing the related checklist parameters of the given disease as validated by the expert;
- TreatmentURI: A system allocated URI pointing to a page containing the disease treatment (it could be in the form of free text, images, videos, etc.) as validated by the expert.

The FARMERS table stores auxiliary information concerning mainly the farmer. More precisely, when a (new) farmer submits a query to the mediator, a tuple is stored in the FARMERS table comprising the following attribute values:

- FarmerID: A system allocated identifier for the (new) farmer; the value of this attribute cannot be NIL;
- CyberbrainID: A system allocated identifier for the person in the Coop who helps/assists/advises the farmer; the value of this attribute cannot be NIL;
- FarmerURI: A system allocated URI pointing to a page containing contact information of the farmer (such as name, mailing address, email address, and phone); the value of this attribute cannot be NIL;
- CyberbrainURI: A system allocated URI pointing to a page containing contact information of the Cyberbrain.

The EXPERTS table is just the list of available experts and the diseases of their expertise. Each entry in this table consists of the following attribute values:

- ExpertURI: contains the expert's contact information (email, phone, etc.); the value of this attribute cannot be NIL;
- DiseaseID: a system allocated identifier for the disease of expertise; the value of this attribute cannot be NIL.

The WEM table contains the results of the Web Extraction Module. When the crawler discovers a (new) disease on the web, a tuple is inserted (automatically) in this table consisting of the following values:

- ExtractionID: A system allocated identifier counting the extraction tasks completed so far; its value cannot be NIL;
- DiseaseURI: This is the URI of the disease discovered by the crawler; its value cannot be NIL;
- Status: Contains the information as to whether an expert has inspected the information contained in DiseaseURI; its value must be one of valid, invalid, non inspected, don't know.

As we shall explain in the next section, experts access the WEM table in order to inspect the URI of newly discovered diseases/treatments. If one or more parts of the inspected URI (diseases, treatments or pictures) are validated by the expert, then the corresponding data is entered in the DISEASES table.

2.2 The Web Extraction Module

The purpose of the Web Extraction Module is to crawl the web in search of new rice diseases and their treatment. It disposes of a table, the WEM table, in which it stores the disease name and the URI pointing to the page where the symptoms and treatment (if any) are described. The WEM table is accessed by the expert to read disease descriptions and perform quality control: if a disease symptom and treatment as appearing in the WEM table are deemed valid then the expert inserts valid under the Status attribute (and the mediator inserts the symptoms and treatment in the DISEASES table); else the expert inserts invalid under the Status attribute. The Web Extraction Module also assists the expert to identify unknown diseases corresponding to farmer queries: if the expert cannot identify a disease based on a farmer query then he can call on the Web Extraction Module, pass to it the farmer query and ask it to search for the disease over the web.

2.3 The Mediator Services

The two main services provided by the mediator are query processing and database maintenance (*i.e.,* updating the tables whenever new knowledge is acquired). However, a number of other services are needed which do not concern individual queries or updates but are related to general collective knowledge regarding rice diseases. For example, based on the contents of the DISEASES table, we can envisage the extraction of association rules correlating symptoms and diseases (assuming that this table contains sufficiently large volumes of data). Moreover, from the analysis of sufficiently large volumes of data, we can setup a dashboard (or observatory) regarding the evolution of diseases across the country, and moreover, notify all farmers if observed symptoms might lead to certain diseases with high probability. These services might require storing additional information in the database, such as weather conditions, type of insects present during the observations by the farmer, and so on. This information can be presented in the form of additional indicators in the checklist that the farmer is asked to complete.

3 The Information Flows

As shown in Figure 1, the information entering the mediator database comes from two sources, the farmer and the expert (directly), as well as from the Web Extraction Module (indirectly, through the expert). Information flow occurs when the mediator interacts with either the farmer or the expert, as well as when the expert interacts with the Web Extraction Module. During these interactions, the database is updated whenever it is appropriate. In this section, we discuss these interactions in some detail as they are important for the system design and implementation.

3.1 Interaction between the Farmer and the Mediator

We recall that when a farmer issues a query to the mediator, the mediator reacts as follows. If the mediator database contains sufficient information so that a single disease can be identified then the mediator returns the appropriate treatment. Otherwise (when no disease or many diseases have been identified), the mediator selects an expert and initiates an interaction directly between the farmer and that expert. In either case, if a disease is identified, the mediator updates the database by:

- entering the disease and its treatment in the `DISEASES` table,
- marking the query status as answered in the `QUERIES` table,
- sending (automatically) a feedback inquiry to the farmer, after a specified period of time has elapsed,
- updating the `QUERIES` table when the feedback of the farmer is received and
- informing the expert if the feedback is unsatisfactory.

Note that, for a given farmer query, the mediator might receive either two or three messages from the farmer, namely: (*i*) the original query from the farmer that consists of one or several photos, (*ii*) the selected photo(s) by the farmer, among those sent by the mediator in response to the previous message, and (*iii*) when the selected photo(s) from the previous message could not allow identifying a single disease, the list of symptoms sent back by the farmer. In each of these steps, the interaction between the farmer and the mediator is the following:

- The mediator performs two tasks: (*a*) it processes the query so as to extract (automatically) specific information items such as the identity of the farmer and the date of issue of the query and updates accordingly the `QUERIES` and `FARMERS` tables; (*b*) based on the photo of the farmer query, the query processor accesses the `DISEASES` table to select similar photos that characterize some diseases. The most similar photos are sent back to the farmer.
- Upon reception of the photos selected by the farmer from the list sent in the previous phase, the query processor accesses again the `DISEASES` table to possibly identify the disease. If a single disease is found then the mediator returns the corresponding treatment. Else, if the photos allow for the identification of *several* diseases, then based on the symptoms of these diseases as stored in the `DISEASES` table, the query processor builds up automatically the list of discriminant symptoms (*i.e.,* those symptoms that are not common to all possible diseases identified so far). This list of symptoms is sent to the farmer as a check list.
- In case a check list has been sent by the mediator, upon reception of the farmer's answer, the query processor accesses the `DISEASES` table to possibly identify the disease, based on this additional information. As in the previous step, if a single disease is found then the mediator returns the corresponding treatment. Else, the mediator selects an expert, sends him/her the farmer query and initiates a direct dialogue between the farmer and the expert. We recall that the choice of an expert by the mediator is based on the information contained in the farmer's query, as well as on information stored in the `DISEASES` and `EXPERTS` tables.

3.2 Interaction between the Expert and the Mediator

As mentioned earlier, the mediator initiates an interaction with an expert when the mediator is unable to process a query issued by a farmer. This happens either when the mediator cannot associate a disease to the query, or when the mediator associates several possible diseases to the query. In either of these cases, the mediator initiates a direct interaction between the farmer and the expert.

3.3 Interaction between the Expert and the Web Extraction Module

The purpose of the Web Extraction Module is to crawl the web in search of new rice diseases and their treatment. The results of this search, namely the URI pointing to the page where the disease and its treatment (if any) are described, are stored in the `WEM` table. The expert initiates an interaction with the Web Extraction Module in two cases:

1. When the expert cannot identify a disease based on a farmer query. In this case, the expert calls on the Web Extraction Module to search for the disease over the web.
2. When the expert is informed of the insertion of a new entry in the `WEM` table. In this case, the expert accesses that table to perform quality control: if the disease description and its treatment are deemed valid then the expert marks the table entry as valid and the mediator inserts a tuple in the `DISEASES` table; else the expert marks the disease description and/or its treatment invalid or don't know.

3.4 Interaction between the Expert and the Farmer

As we mentioned earlier, when the mediator cannot identify a single disease based on the farmer query, then the mediator selects an expert and initiates an interaction between the farmer and that expert. If the disease is identified during the interaction between the farmer and the expert, and it is a new disease, then the expert initiates an interaction with the mediator so that the (new) disease and the appropriate treatment (if any) are inserted in the `DISEASES` table of the database. During this interaction, the mediator performs two tasks: (*a*) inserts the disease description and the disease treatment in the `DISEASES` table, (*b*) if a treatment has been sent to the farmer then the mediator marks the query as answered in the `QUERIES` table (using the `QStatus` attribute), and after some specified period of time sends a feedback request to the farmer.

4 Concluding Remarks

We have seen an approach to the diagnosis and treatment of rice diseases and outlined the architecture of a computer aided system tailored to this problem. The system is currently under development as part of a Franco-Thai project.

The heart of the system is a software mediator allowing farmers to formulate distant queries regarding a disease and to receive recommendations for a treatment of the disease if one exists. The disease diagnosis and the recommended treatment are based on expert knowledge stored in the mediator database. The processing of farmer queries may involve direct access to the mediator by farmers, online communication between the mediator and the experts, or a direct dialogue between the farmer and an expert.

The main features of our approach are the following: (1) It requires a minimal effort on the part of the farmer (*i.e.,* completing a checklist of symptoms plus taking a few pictures of diseased parts of rice plants; and (2) The system does not replace the expert but simply assists the expert whenever the stored expertise can answer the farmer query automatically.

Future work aims mainly at providing the necessary services for planning on a national scale. On the other hand, it should be clear that our information model is generic in the sense that it can be used to support the needs of farmers in a variety of similar environments, with minor changes. More generally, our information model applies for the design of any decision making system involving interactions between users and experts relying on multimedia documents and on information stored in an associated database. Health care applications dealing with distant diagnosing are examples of such applications. The design of a generic information model of which the system presented in this paper would simply be an instance is also part of our future work.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston (1999)
2. Caracciolo, C., Stellato, A., Rajbahndari, S., Morshed, A., Johannsen, G., Keizer, J., Jaques, Y.: Thesaurus maintenance, alignment and publication as linked data: the agrovoc use case. Int. J. Metadata Semant. Ontologies 7(1), 65–75 (2012)
3. Kawtrakul, A., Mulasastra, I., Khampachua, T., Ruengittinun, S.: The challenges of accelerating connected government and beyond: Thailand perspectives. Electronic Journal of e-Government 9(2), 183–202 (2011)
4. Kawtrakul, A., Sriswasdi, W., Wuttilerdcharoenwong, S., Khunthong, V., Andres, F.: Cyberbrain: Towards the next generation social intelligence. In: World Conference on Agricultural Information and IT, pp. 1–8 (2008)

Information Visualization for Chronic Patient's Data

Shuichi Toyoda[1] and Noboru Niki[2]

[1] Department of Nursing, Jobu University, Japan
toyoda@jobu.ac.jp
[2] Department of Optical Science and Technology, University of Tokushima, Japan
niki@opt.tokushima-u.ac.jp

Abstract. Medical data are generated in large quantities every day. There are many aspects to medical data, including clinical information, administration data, and time granularity, and the number of chronic disease patients increases yearly. However, clinicians have limited time to review and process patient data. Information visualization is therefore required for the efficient management and utilization of the data. The management of chronic disease requires information technology if it is to improve the quality and efficiency of health care. In this paper, we consider the visualization of medical data, focusing on the diversity of medical data and chronic disease care.

1 Introduction

Accelerating digitization in the medical field in recent years has resulted in the daily production of large quantities of medical data, which is often beyond the recognition and management capabilities of the user [1].

Increasingly, clinicians and medical staff want to identify new information from the accumulated data and utilize it to make a reliable interpretation or summary of a case. However, in the highly specialized domain such as medicine, interpretation of the data may vary greatly in accordance with the knowledge and experience of the individual user [2]. Providing relevant information for users is therefore important, and the application of data-mining and information visualization techniques to the medical field is expected to prove useful [3]. In addition, increases in the number of chronic disease patients have increased medical costs, and efficient chronic disease care is becoming an important consideration in the reduction of these costs.

In this paper, we consider the visualization of medical data, focusing on the diversity of medical data and chronic disease care.

2 Characteristics of Medical Data

In medical data, the diversity is one of characteristics. This chapter briefly describes clinical data, medical knowledge, administration data, user types, and time granularity. Figure 1 shows the diversity of medical data.

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 81–90, 2013.

Clinical data include basic patient information, observations, order data, laboratory test results, physiological tests, and medical images. Drug- and test-ordering data are common types of order data. Blood and urine tests are typical laboratory tests. For laboratory results, transitions in, and periodicity of, data are important, in addition to the individual results [4]. Computed tomography (CT) data and Magnetic Resonance Imaging (MRI) data are typical medical images. An example of a typical physiological test is the electrocardiogram (ECG).

Medical knowledge and administration data include care processes, clinical guidelines, and clinical pathways. A care process shows the progress of the medical treatment of the disease for a patient. Computerization of diagnosis and medical treatment process data has advanced rapidly with the digitization of medical information. Analyzing the data on the diagnosis and treatment in the care process must take account of the diversity of contents and time granularity, and the complexity of data correlation. Therefore, data analysis that includes a diagnosis and treatment process will require substantial data and time.

The American Institute of Medicine has defined clinical guidelines as ‘systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances’ [5]. By using clinical guidelines, it is possible to improve the quality of medical care and reduce the medical cost.

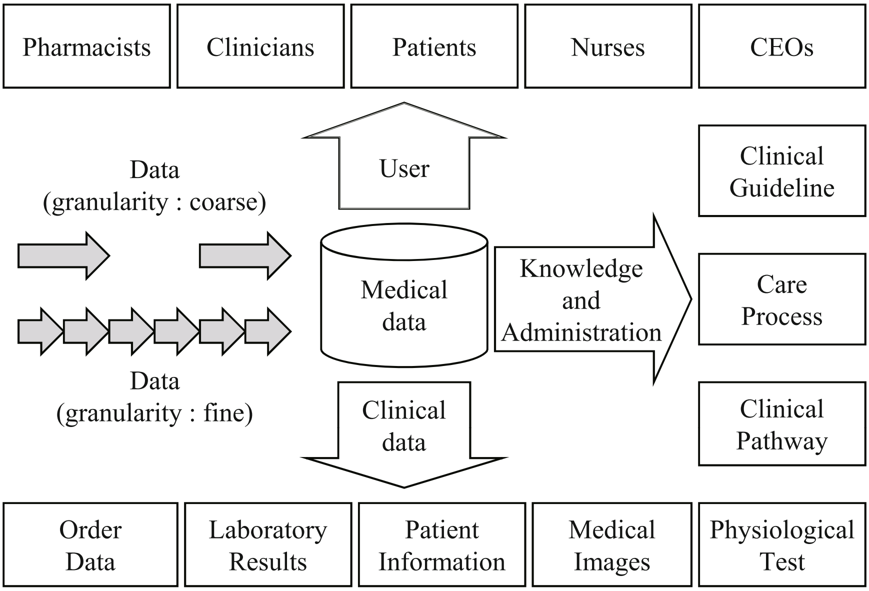


Fig. 1. The diversity of medical data: Clinical data, administration data, and user type

There are many types of users, including patients, clinicians, nurses, pharmacists, rehabilitation staff, executive staff such as CEOs, and health maintenance organizations. Users expect to have access to information relevant to their purposes. It is necessary for clinicians to store large amounts of patient records, but it is not easy for them to extract relevant information from the large amount of data. They need to access a particular subset of patient records according to their specific focus. In contrast, CEOs seek the best practice model or clinical pathways to improve the quality of medical care and to contain costs. A sophisticated analysis of the diagnosis and treatment process is needed for the development and management of the clinical pathway [6].

The time granularity in patient records varies from fine to coarse. Time granularity is considered in accordance with the type of medical service. Examples are patient-centric acute-patient medical services, and disease-centric chronic-patient medical services. The appropriate level of time granularity for acute patients is fine, such as by hour or minute. The pulse and blood pressure of a patient, and a newborn baby's heartbeat may be measured with a fine granularity at high frequency [7]. Conversely, the level of time granularity for chronic patients can be coarser, such as by month or year. The blood glucose level in diabetic patients is data of a coarser granularity as it is recorded monthly (i.e., at low frequency).

3 Chronic Disease Care

In this chapter, we describe chronic disease care, in which the use of information technology can play an important role.

Chronic disease care can continue for long periods. In addition, patients with chronic diseases, such as diabetes and heart diseases, often have more than one disease. Care is necessary to ensure that the patient condition does not get worse, and complication prevention is also important [8]. In the care of the chronic disease patient, it is necessary to provide diagnosis and treatment that corresponds to the patient's health conditions. The role of the clinical information system in chronic disease care management is therefore important, as are self-management support, decision support, and delivery system redesign [9]. Delivery system redesign refers to the use of care management and planned visits.

During chronic disease care, types of test data are produced, such as tests to check the chronic disease state, tests for the appearance of complications, and tests to check other symptoms. For example, the management of a diabetes patient will include blood glucose and HbA1c testing, lipid testing, renal function testing, eye examination, and chest X-ray examinations. The tests are carried out at various intervals, so when preparing the test plan it is important to make it easy to understand the temporal context information, such as the frequency and the relationship between the tests. The test plan description should be in a table-based form rather than a free text form. The table-based form description facilitates the review and improvement of the test plan, and has the advantage of reusing the accumulated data [10].

The drug-ordering data may involve a mixture of drugs to be used continuously for a long period for chronic diseases, drugs to be given for treatment of a specific disease for a certain period, and other drugs. Therefore, low-frequency drug-ordering data could be buried in the high-frequency routine data and is at a high risk of being overlooked.

Information systems, such as clinical reminder systems and medication treatment systems, are applied to chronic disease management. Sequist et al. conducted research on clinical reminder systems [11]. This study involved clinical reminders based on clinical guidelines. For diabetes patients, five types of reminders, such as a cholesterol test and an HbA1c test, are issued. Koutkias et al. proposed a personalized framework for medication treatment [12]. This framework supports self-management of chronic disease patients, is applied to hypertension management, and focuses on adverse drug events. Its design is based on service-oriented architecture and it is implemented by smartphone.

4 Information Visualization

Information visualization is one of the technologies used to solve the problems associated with handling large amounts of information. It not only makes the exploration of information easy, but also contributes to the increased information awareness of the user. Consider the huge quantity of patient data that clinicians now face. The time available for the clinician to refer to and retrieve patient data is limited. Therefore, they now expect to have a system with functions that enable easy access to valuable data during their decision-making, i.e., a specific subset of patient data. One of the techniques aimed at solving this problem is information visualization.

To access medical data from a specific viewpoint, the requirement of domain-specific knowledge must be considered. This means that the information visualization of medical data should be supported by a domain-specific knowledge base.

The use of information visualization in the medical field can be classified into two categories, depending on its purpose. One category involves visualization to support information awareness. It allows users to understand the contextual information, and is promoted by information reorganization and multiple viewpoints [13]. Medical data can be reorganized from a variety of viewpoints such as chronological, comparative, and summarized views, and an overview [14]. Additionally, the multiple viewpoints can support different perspectives and can help users understand complex relationships among data sets. Visualization to support information awareness makes use of the user's background expertise. An appropriate visualization method will depend on the characteristics and structure of the object data.

The second category is visualization to support information exploration. It is intended to assist the selection of the scope of exploration by visual adaption and by representing the exploration results via an interactive information-exploration system. This visualization is often used with data mining and parallel coordinates. Applying data mining, a visual operation pathway such as concept hierarchy might be offered [15]. Applying parallel coordinates, multidimensional data sets are transferred into two-dimensional pattern data [16]. For example, the data set of blood sugar values for each diabetic patient might be expressed by two-dimensional patterns, and pattern data will be analyzed. In a visual analysis, the operation can easily be repeated until the user is satisfied with the retrieval result [17].

Moreover, one characteristic of the medical domain is the great variation in domain knowledge among users. For example, in information representation for chronic disease patients, the system intended for the doctor should display various information on a single screen, such as the presence of unexpected values, the presence of complications, recent laboratory results, and the treatment plan. The system intended for the patient and their family should display information such as the meaning of the laboratory results, methods to prevent complications, and the importance of compliance with recommendations for medication. In addition, it is necessary to note the following characteristics. Because the data provides information relating to individuals, consideration should be given to preserving anonymity. Moreover, the data collected in databases is often incomplete if it sees from the aspect of analysis [18].

5 Visualization System for Medical Data

In this chapter, we describe examples of visualization systems for the medical field. Table 1 compares system features.

5.1 Information Awareness Type

Visualization systems that support the information awareness include the systems intended for clinical guidelines and personal medical histories.

AsbruView is a visualization system for clinical guidelines [19]. It deals with treatment plans expressed in the Asbru language. Complicated structures that include temporal uncertainties in the medical treatment plan, continuance conditions, and forced or optional enforcement are visualized to ensure the accurate selection and improvement of the medical treatment plan. AsbruView adopts iconic technology. It expresses medical treatment plans from two viewpoints; namely as temporal view and as topological view. Changes in value can be expressed by changing the form, size, and color of the graphical object.

LifeLines is a system that expresses personal medical histories, including symptoms and treatment, at a coarse time granularity [20]. LifeLines visualizes medical histories in terms of Gantt charts and uses color to group data semantically [21].

5.2 Information Exploration Type

The visualization system that supports information exploration interactively includes the visual search system of test results using the parallel coordinate and the analysis system for care process using data mining.

Klimov et al. reported on the user-driven software, Temporal Association Charts (TACs), which has interactive knowledge-based visualization functions [22]. The TACs enable users to explore and graphically analyze the time and value associations among domain concepts. The TACs aggregate patients' laboratory test data by temporal abstraction [23]. The system uses a knowledge base to prepare a time summary of laboratory test data, seeking observations that consolidate two or more patients' clinical data in the analysis of a large patient population. The TACs adopt parallel coordinates as visualization techniques and can display two or more patients' data at the same time. By mapping two or more test items on the axes of parallel coordinates and by using variety in the width and color attributes of the polygonal line connecting the axes, various relationships can be represented.

Bito et al. report on a system, hereinafter referred to as VisDRG, which analyzes the relationship between the care process and the cost by applying data warehouse technology [24]. VisDRG is an interactive mechanism for supporting the analysis of complicated relationships in a large quantity of inpatient care-process data. It uses professional expertise for a dynamic classification of the patients and medical services. VisDRG contributes to the analysis and understanding of the components of care and cost by analyzing the progress of the care process for each diagnosis group (DRG refers to the group to which the inpatient's medical-fee payment belongs). In VisDRG, a three-dimensional data cube with axes for patients, medical services, and time is constructed. An interactive information-visualization method that displays the care-process pattern, correlation, and quantitative data in the bar charts and as a two-dimensional matrix is realized, while operations on the cube select the patient group and service group to be analyzed..

Table 1. Comparison of visualization system for medical data

System	Category	Data	Information Technology	Time Granularity
AsbruView	Awareness	Clinical guideline	Graphic, Iconic	Coarse
LifeLines	Awareness	Medical history	Gantt chart	Coarse
TACs	Exploration	Laboratory test	Parallel coordinate	Fine
VisDRG	Exploration	Care process	Cube operation	Medium
SAKURA-Viewer	Awareness	Order history	Concept hierarchy	Medium

6 Visualization for Chronic Disease Care

Chronic disease management requires communication between patients and the health care provider. We consider information visualization that contributes to chronic disease care from the two aspects of clinician and patient.

Clinicians are experts with specialized knowledge of the medical domain. They can perceive large patterns of meaningful information in the medical domain, and their time is limited. Visualization for clinicians requires the effective utilization of the screen area and reduction of information redundancy. Clinicians need representation from the specialized viewpoint of chronic disease.

Patients are novices with only common sense or everyday knowledge of the medical domain. Visualization for patients is often required for the promotion of self-management of chronic disease care. Patients need structured visual information to understand the conditions of their disease and the educational issues.

Here, we introduce SAKURA-Viewer and FUJI-Viewer, which we have studied and developed for clinicians. These viewers have been developed to deal with order data for chronic disease patients. Figure 2 shows an overview of these viewers.

6.1 SAKURA-Viewer

SAKURA-Viewer is a visualization mechanism that makes it easy to keep track of the order history by efficiently displaying the drug- and test-order histories of each individual patient [25]. It contributes to understanding the diversity and continuity of order data with regard to the doctor's office system.

SAKURA-Viewer visualizes order history from temporal and semantic viewpoints simultaneously. These viewpoints are extracted from order attributes. The analysis of order attributes is based on concept hierarchy. The temporal viewpoint is a conceptual high-level attribute set and the semantic viewpoint is a conceptual low-level attribute set. SAKURA-Viewer adopts button objects as the basic data display unit. This feature realizes a data entry support system that enables the displayed data to be used directly as input data. It uses color attributes, such as red for execution-attention items and sepia for adoption-cancelled items, to facilitate the recognition of attributes.

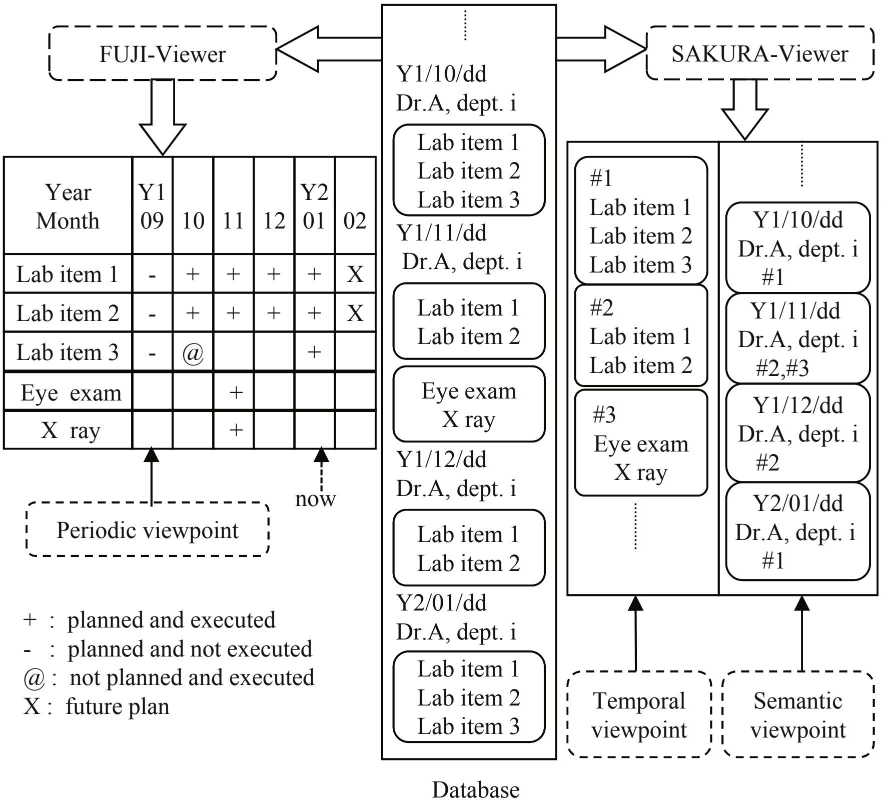
The evaluation of SAKURA-Viewer was performed using the order data for approximately 3300 chronic disease patients during a nine-month period. The results showed that the quantity of drug-order items displayed was compressed down to approximately 20% of that required for a chronological display. SAKURA-Viewer is an effective tool for clinicians to check the order history that involves the care of chronic disease patients.

6.2 FUJI-Viewer

FUJI-Viewer is a visualization mechanism that presents the periodicity of test order data and test plan data [26]. This viewer enables comprehension of the diversity of the intervals among test items. This viewer visually supports the understanding of the difference between a plan and its execution.

FUJI-Viewer visualizes test order histories and test plans from a periodic viewpoint. This viewer displays test order history on a monthly tab. It represents plan information overlapping test plan data simultaneously in a single tab, and represents the relationship between test plan information and test execution information on a monthly basis. Moreover, a function enabling the reuse of previous test plans and test-execution information in the construction of a future test plan has been realized. By using this function, the whole body control of patients is supported, and the planning burden is reduced.

The evaluation of FUJI-Viewer was performed using the data collected from 20 patients during one year. The results showed that 81% of test plans were carried out as intended, and a high compliance rate was observed for the test plans of chronic disease patients. FUJI-Viewer succeeded in improving the efficiency of the continuous diagnosis and treatment required for chronic disease patients.



Year Month	Y1 09	10	11	12	Y2 01	02
Lab item 1	-	+	+	+	+	X
Lab item 2	-	+	+	+	+	X
Lab item 3	-	@			+	
Eye exam			+			
X ray			+			

+ :	planned and executed
- :	planned and not executed
@ :	not planned and executed
X :	future plan

Fig. 2. Overview of SAKURA-Viewer and FUJI-Viewer

7 Conclusion

In this paper, several types of information visualization for chronic patients' data are investigated, and the use of visual information is described by classifying it according to the twin viewpoints of information awareness and information exploration. Medical data have enormous volume, but involve complex structures and diverse relationships. Most of the medical data generated in large quantities every day at the medical frontline are used only for the diagnosis and treatment of the patient concerned. We consider that information visualization can play a significant role in reusing patient diagnosis and treatment data that would otherwise be unused.

References

1. Weiss, G.: Welcome to the (almost) digital hospital. IEEE Spectrum 39(3), 44–49 (2002)
2. Patel, V.L., Kaufman, D.R.: Cognitive Science and Biomedical Informatics. In: Biomedical Informatics: Computer Applications in Health Care and Biomedicine, pp. 133–185. Springer (2006)

3. Card, S.K., Mackinlay, J.D., Shneiderman, B.: Information Visualization. In: Readings in Information Visualization Using Vision to Think, pp. 1–34. Morgan Kaufmann (1999)
4. Bauer, D.T., Guerlain, S., Brown, P.J.: The design and evaluation of a graphical display for laboratory data. Journal of the American Medical Informatics Association 17(4), 416–424 (2010)
5. Institute of Medicine: Clinical practice guidelines: Directions for a new program. National Academy Press, Washington, DC (1990)
6. Chu, S., Cesnik, B.: Improving clinical pathway design: lessons learned from a computerized prototype. International Journal of Medical Informatics 1(51), 1–11 (1998)
7. Venkatasubramanian, K.K., Gupta, S.K.S., Jetley,R.P., Jones, P.L.: Interoperable Medical Devices. IEEE Pulse, 16–27 (September/October 2010)
8. Bodenheimer, T.: Interventions to Improve Chronic Illness Care: Evaluating Their Effectiveness. Disease Management 6(2), 63–71 (2003)
9. Osheroff, J., Pifer, E.A., Teich, J.M., Sittig, D.F., Jenders, R.A.: Improving Outcomes with Clinical Decision Support: An Implementer's Guide. Himss (2005)
10. Institute of Medicine: Applying Evidence to Health Care Delivery. In: Crossing the Quality Chasm: A New Health System for the 21st Century, pp.145–163. National Academy Press, Washington, DC (2001)
11. Sequist, T.D., Gandhi, T.K., Karson, A.S., Fiskio, J.M., Bugbee, D., Sperling, M., Cook, E.F., John Orav, E.J., Fairchild, D.G., Bates, D.W.: A Randomized Trial of Electronic Clinical Reminders to Improve Quality of Care for Diabetes and Coronary Artery Disease. Journal of the American Medical Informatics Association 4(12), 431–437 (2005)
12. Koutkias, V.G., Chouvarda, I., Triantafyllidis, A., Malousi, A., Giaglis, D., Maglaveras, N.: A Personalized Framework for Medication Treatment Management in Chronic Care. IEEE Transactions of Information Technology in Biomedicine 2(14), 464–472 (2010)
13. Baldonado, M.Q.W., Woodruff, A., Kuchinasky, A.: Guidelines for Using Multiple Views in Information Visualization. In: Proceedings of the 5th International Working Conference on Advanced Visual Interfaces, pp.110–119 (2000)
14. Bui, A.A.T., Aberle, D.R., Kangarloo, H.: TimeLine: Visualizing Integrated Patient Records. IEEE Transactions of Information Technology in Biomedicine 4(11), 462–473 (2007)
15. Han, J., Kamber, M.: Data Mining Concepts and Techniques. Morgan Kaufmann Publishers (2001)
16. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: IEEE Proceedings of the 1st Conference on Visualization, Los Alamitos, pp. 361–378 (1990)
17. Starren, J., Johnson, S.B.: An object-oriented Taxonomy of Medical Data Presentations. Journal of the American Medical Informatics Association 1(7), 1–20 (2000)
18. Cois, K.J., Moore, G.W.: Uniqueness of medical data mining. Artificial Intelligence in Medicine 26(1-2), 1–24 (2002)
19. Kosara, R., Miksch, S.: Metaphors of movement: a visualization and user interface for time-oriented, skeletal plans. Artificial Intelligence in Medicine 2(22), 111–131 (2001)
20. Plaisant, C., Milash, B., Rose, A., Widoff, S., Shneiderman, B.: LifeLines: Visualizing Personal Histories. In: Proceedings of the ACM Conference on Human Factors in Computing Systems, pp. 221–227 (1996)
21. Ware, C.: Color. In: Information Visualization: Perception for Design, pp. 103–149. Morgan Kaufmann, San Mateo (2000)
22. Klimov, D., Shahar, Y., Taieb-Maimon, M.: Intelligent Interactive Visual Exploration of Temporal Associations among Multiple Time-oriented Patient Records. Methods of Information in Medicine 3(48), 254–262 (2009)

23. Shaher, Y.: A framework for knowledge-based temporal abstraction. Artificial Intelligence 90, 79–133 (1997)
24. Bito, Y., Kero, R., Matsuo, H., Shintani, Y., Silver, M.: Interactively Visualizing Data Warehouses. Journal of Healthcare Information Management 2(15), 133–142 (2001)
25. Toyoda, S., Niki, N., Nishitani, H.: SAKURA-Viewer: Intelligent Order History Viewer based on Two-Viewpoint Architecture. IEEE Transactions of Information Technology in Biomedicine 2(11), 141–152 (2007)
26. Toyoda, S., Niki, N., Nishitani, H.: A Test-Data Management Function for Chronic Condition Patients. In: Proceedings of the 20th IEEE International Symposium on Computer-Based Medical Systems, pp. 707–712 (2007)

Exploiting Semantic and Social Information in Recommendation Algorithms

Dalia Sulieman[1,2], Maria Malek[2], Hubert Kadima[2], and Dominique Laurent[1]

[1] ETIS-ENSEA Université de Cergy-Pontoise CNRS - France
[2] EISTI Ecole Internationale des Sciences du Traitement de l'Information - France
{dalia.sulieman,maria.malek,hubert.kadima}@eisti.eu,
dominique.laurent@u-cergy.fr

Abstract. In this paper we present algorithms for recommender systems. Our algorithms rely on a semantic relevance measure and a social network analysis measure to *partially* explore the network using depth-first search and breath-first search strategies. We apply these algorithms to a real data set and we compare them with item-based collaborative filtering and hybrid recommendation algorithms. Our experiments show that our algorithms outperform existing recommendation algorithms, while providing good precision and F-measure results.

1 Introduction

In order to cope with the huge increase of information available on the Web, *recommender systems* are widely used to help users in making choices according to their interests. Furthermore, social networks provide relevant information for social recommender systems [1]. Unfortunately, due to the size of these social networks, the design of efficient algorithms for collecting this information remains an important open issue.

In this paper we introduce a *semantic social recommender system*, in which we suppose given a set of users and a set of items such that users are connected through a social network, and users and items are described via an ontology. In this setting, given an item, we propose heuristic based search algorithms to explore the social network *as few as possible* in order to compute a relevant set of users to whom the item can be recommended. Our main contribution is to provide algorithms that combine all available information (the domain ontology relating all items, user profiles seen as part of that ontology and the social network connecting users) in order to efficiently compute the relevant set of users. Our proposed algorithms have been implemented and tested against a real data set (Amazon data set). Our experiments show that our algorithms outperform existing approaches (item-based collaborative filtering and hybrid recommendation algorithms) while providing good precision and F-measure results.

The rest of the paper is organized as follows: In Section 2 we describe the semantic and social information used in our approach. Algorithms are the subject of Section 3 and in Section 4 we report on experiments conducted on a real data set. In Section 5 we briefly review related work and Section 6 contains concluding remarks and suggestions for future work.

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 91–100, 2013.

2 Semantic and Social Information

The semantic information generally relies on three fundamental aspects:

1. *User profiles* that can be represented in several ways (e.g. vectorial representation or conceptual representation [2]).
2. *Domain ontology* generally organized as a tree in order to describe objects of a given domain. Several recommender systems use ontology to estimate users profiles, in the case of lack of information about users [3].
3. *Semantic similarity measures* used to compute the relevance between ontology concepts [4].

In our approach we use a domain ontology to represent the knowledge about users and items. We attach a *semantic-ontology profile* to each user and item, and we use a *semantic similarity measure* to compute the semantic relevance between users and items. For that, we introduce the following definitions.

Definition 1. *Given a set of items, the Semantic Ontology Tree* (SOT) *associated with these items is a tree whose nodes are pairs of the form* (τ, λ)*, where* τ *is a term of the considered ontology and* λ *is an integer indicating the level in the* SOT *where* τ *occurs. Moreover, we assume that every item is associated with a* unique *leaf of the* SOT*.*

Given an SOT, we associate every item x with a subset of this ontology. This subset, called *Item Profile Tree of* x and denoted by $IPT(x)$, is the path of the SOT containing all pairs (τ, λ) connecting the root of the SOT with the leaf to which item x is associated.

In our model, we suppose that every user u is associated with a set of items, denoted by $I(u)$, containing all items u bought and liked in the past. Based on this information, we define the *user profile tree* of a given user as follows.

Definition 2. *Let* u *be a user and* $I(u)$ *its associated set of items. The* User Profile Tree of u*, denoted by* $UPT(u)$*, is the union of all item profile trees of the items in* $I(u)$*. In other words, we have:* $UPT(u) = \bigcup_{x \in I(u)} IPT(x)$*.*

Figure 1, shows the IPTs of items $Item1$, $Item2$ and $Item3$, and the UPT associated to a user who bought and liked these items.

Based on the previous definitions, we present a user-item semantic relevance measure between users and items. To do so, we introduce a similarity measure between two profile trees P_1 and P_2, denoted by $\sigma(P_1, P_2)$, as follows:

$$\sigma(P_1, P_2) = \frac{1}{\mu}\left(\sum_{(\tau,\lambda)\in P_1 \cap P_2} \lambda\right)$$

where $\mu = \min\left(\sum_{(\tau,\lambda)\in P_1} \lambda, \sum_{(\tau,\lambda)\in P_2} \lambda\right)$. Using σ, we now define the similarity between a user u and an item x as the similarity between their profile trees.

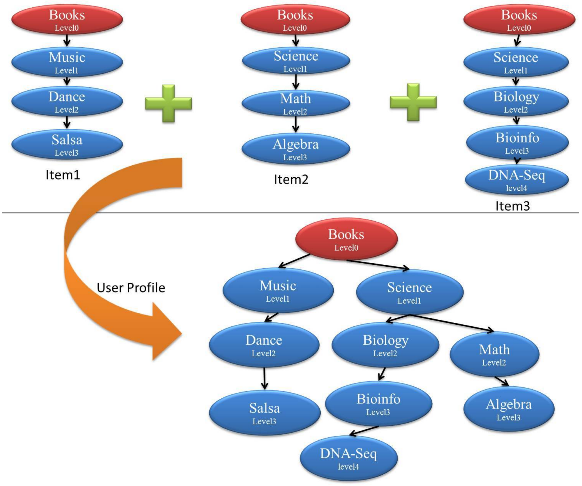


Fig. 1. A User Profile Tree

Definition 3. *Let u be a user and x an item. The similarity measure between u and x, denoted by $sim(u, x)$, is $sim(u, x) = \sigma(UPT(u), IPT(x))$.*

As can be seen from Definition 3, sim is not a standard similarity measure since it applies to arguments of different types (namely a user and an item). In fact, $sim(u, x)$ measures to which extent items 'similar' to x can be found in $UTP(u)$. Notice in this respect that if u bought and liked x, then $ITP(x)$ is a subtree of $UTP(u)$, in which case $sim(u, x)$ is maximal, that is, equal to 1.

Example 1. Figure 1 shows an example of user u, who likes items $Item1$, $Item2$ and $Item3$. In this case, we have $I(u) = \{Item1, Item2, Item3\}$ and $UPT(u) = IPT(Item1) \cup IPT(Item2) \cup IPT(Item3)$. For item x such that $IPT(x) = \{(Books, 0), (Science, 1), (Math, 2), (Geometry, 3)\})$, $sim(u, x)$ is computed as follows: since $UPT(u) \cap IPT(x) = \{(Books, 0), (Science, 1), (Math, 2)\}$, $\sum_{(\tau,\lambda)\in UPT(u)} \lambda = 21$ and $\sum_{(\tau,\lambda)\in IPT(x)} \lambda = 6$, we have $sim(u, x) = 3/6 = 0.5$.

In the algorithms to be presented next, the similarity measure sim is used to discard users whose relevance with the item to be recommended is too low.

The second component of our model is the Social Information, which is mainly based on *social networks* [5], and *degree centrality* [6].

Collaboration social networks are generally extracted from bipartite graphs, using the one-mode projection [5,7]. In our approach, we assume that we are given a bipartite graph whose edges connect users to items. Based on such a bipartite graph, we consider a user one-mode projection whose nodes are users and whose edges are weighted by the number of products the connected users have bought and liked in common.

This graph, seen as a *social network*, is explored by our algorithms in order to build up a list of users to whom a given item can be recommended. In order to explore the relevant part of this graph, *degree centrality* of nodes is one of the most popular social network analysis measures [6]. If v is a node of a non directed graph, the degree centrality of v is defined as the number of edges involving this node [7]. Our algorithms use the degree centrality as a measure to guide the search in the social network.

3 Semantic-Social Recommendation Algorithms

We recall that, in our approach, the graph to be searched is a social network in which nodes are users and edges are weighted in order to take into account the number of items two connected users bought and liked. Denoting this graph by G, and in order to avoid exploring *all* nodes and *all* edges, we apply heuristics depending on semantic similarity, degree centrality and edge weights. Moreover, these heuristics are combined with two distinct strategies for graph traversal, namely depth-first search, referred to as SSDFS in Algorithm 2 and breadth-first search, referred to as SSBFS in Algorithm 3. Depending on the chosen search strategy, one of these algorithms is called by Algorithm 1 (line 6).

In order to avoid visiting all nodes and edges of G, while visiting as many relevant nodes as possible, the exploration of G starts through the nodes that have a high degree centrality, that is, the nodes that are connected to a high number of other nodes. To do so, the N nodes having the highest degree centrality are computed and stored in a vector called Top-N-vector (lines 4-5 of Algorithm 1). When exploring G, the only paths going through a node from this vector are considered (line 1 of Algorithm 2 and line 2 of Algorithm 3). Starting form the nodes in Top-N-vector, the graph is explored either in a depth first manner (see Algorithm 2) or in a breadth-first manner (Algorithm 3).

Algorithm 1. Semantic-Social main algorithm

```
Require: ($i$) A user-item bipartite graph $G$ with nodes $V$ and edges $E$
    ($ii$) An item $x$ and its Item Profile Tree $IPT(x)$
    ($iii$) A positive integer $N$
    ($iv$) A user-item similarity threshold $\theta$
    ($v$) An edge weight threshold $\delta$
Ensure: List of recommended users $user\_list$
 1: for all nodes $v$ in $G$ do
 2:     $v.label = unvisited$
 3: $user\_list$ = empty list
 4: Compute the degree centrality of every node in $G$
 5: Top-$N$-vector = all nodes of $G$ with the top-$N$ highest degrees centrality
 6: Call one of the following search algorithms (SSDFS or SSBFS) with input Top-$N$-
    vector, $x$, $IPT(x)$, $\theta$ and $\delta$
 7: return $user\_list$
```

In the case of depth-first search, every node v from Top-N-vector is processed if v is unvisited, and if its similarity with item x is greater than the threshold θ. Then, all successors v' of v are recursively processed in the same way (line 2 of Algorithm 2), until reaching a node that fails to satisfy the similarity requirement or until reaching an edge whose weight is less than the threshold δ.

Algorithm 2. Depth First Search algorithm (SSDFS)

```
 1: for all $v$ in Top-$N$-vector do
 2:   Call Recursive-Search with input $v$

 3: Function Recursive-Search
 4: if $v.label = unvisited$ then
 5:   $v.label = visited$
 6:   if $sim(v, x) > \theta$ then
 7:     Add $v$ to the current value of $user\_list$
 8:     for all edge $e = (v, v')$ in $G$ do
 9:       if $e.weight > \delta$ then
10:         Call Recursive-Search with input $v'$
11: End Function
```

In the case of a breadth-first search strategy, all nodes from Top-N-vector whose similarity with the input item x is greater than θ are first inserted into a queue Q and added to the output list (lines 2-6 of Algorithm 3). Then while Q is not empty, starting from the first node v in Q, all successors v' of v whose connection with v has a weight greater than δ are processed as follows: if v' is unvisited and if $sim(v', x) > \theta$, then v' is added to the output list and v' is inserted into Q (lines 9-14 of Algorithm 3).

Algorithm 3. Breadth First Search Algorithm (SSBFS)

```
 1: $Q$ = empty queue
 2: for all $v$ in Top-$N$-vector do
 3:   $v.label = visited$
 4:   if $sim(v, x) > \theta$ then
 5:     Add $v$ to the current value of $user\_list$
 6:     $enqueue(Q, v)$
 7: while $Q \neq \emptyset$ do
 8:   $v = deque(Q)$
 9:   for all edge $e = (v, v')$ in $G$ do
10:     if $v'.label = unvisited$ and $e.weight > \delta$ then
11:       $v'.label = visited$
12:       if $sim(v', x) > \theta$ then
13:         Add $v'$ to the current value of $user\_list$
14:         $enqueue(Q, v')$
```

4 Experiments and Results

In our experiments, we have used *Amazon* data (whose description is available at http://snap.stanford.edu/data/amazon-meta.html) as a real dataset to test our algorithms and compare them with existing other approaches. This dataset contains information about users and their previous purchases along with information about items, in the form of an ontology. The social network obtained from these data is a graph containing 38, 982 nodes and more than 5 million edges. Algorithms are assessed according to the following criteria:

- *Accuracy measures:* precision, recall and F-measure [8]. *Precision* is computed by dividing the number of users who are in the recommendation list and who bought the item (true positives), by the number of all users in the recommendation list (true positives + false positives). *Recall* is computed by dividing the number of users who are in the recommendation list and who bought the item (true positives), by the number of all the users who bought the item, not necessarily in the recommendation list (true positives + false negatives). *F-measure* is defined by $2 * \frac{Precision*Recall}{Precision+Recall}$.
- *Graph coverage:* percentage of visited nodes.
- *Runtime:* time taken to answer a recommendation query.

We first report on the impact on F-measure and on the percentage of visited nodes of the parameters θ, δ and N. Based on these experiments, whose results are depicted in Figures 2 to 4, the values for θ, δ and N have been respectively set to 3, 40 and 200, for further experiments.

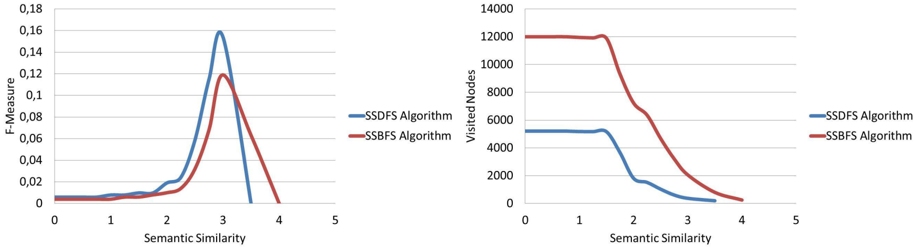


Fig. 2. F-Measure and graph visited nodes according to θ

We compare our algorithms with two of the most common recommendation algorithms, namely (*i*) *Collaborative filtering* algorithm (CF), using cosine similarity measure to compute the similarity between two items [9], and (*ii*) *Hybrid recommendation* algorithm (HR), which combines collaborative filtering recommendation using cosine similarity measure, and content-based recommendation using our semantic-similarity measure *sim*.

The four recommendation algorithms (SSDFS, SSBFS, CF and HR) have been run for 58 recommendation queries and the average precision, recall, F-Measure, runtime, and graph coverage have been computed. The obtained results are

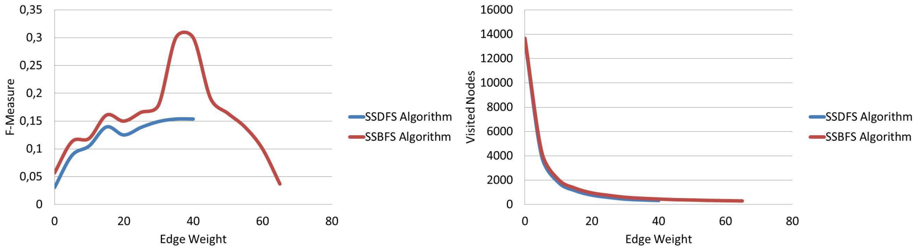


Fig. 3. F-Measure and graph visited nodes according to δ

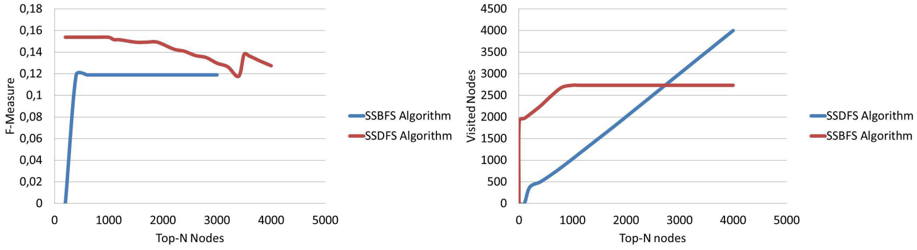


Fig. 4. F-Measure and graph visited nodes according to N

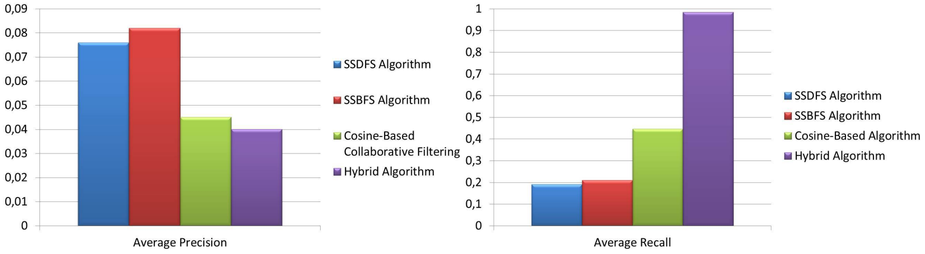


Fig. 5. Comparison of precision and recall between SSDFS, SSBFS, CF and HR

shown in Figures 5 to 7. Figure 5 shows that SSDFS gives the best precision and that HR gives the best recall, while Figure 6 shows that SSDFS gives the best F-Measure. Moreover, Figure 7 clearly shows that the performance and graph coverage of SSDFS and SSBFS outperform the performance and graph coverage of CF and HS (according to which *all* nodes are visited).

We notice that SSDFS and SSBFS have better performance and F-measure than CF and HR, because CF and HR algorithms are designed to explore the whole dataset, while SSDFS and SSBFS are designed to explore only a part of the dataset. More precisely, SSDFS and SSBFS start the search from the node whose degree centrality is the highest, so if user-item relevancy is high, these algorithms will suppose that the connections have a high relevancy too. Then, the number of visited nodes and runtime are reduced, while the precision is improved. On the other hand, the fact that SSBFS has a better F-measure than SSDFS can be explained as follows: if the user-item relevancy is high for a node,

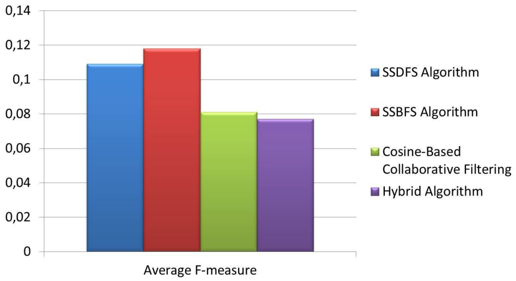


Fig. 6. Comparison of F-Measure between SSDFS, SSBFS, CF and HR

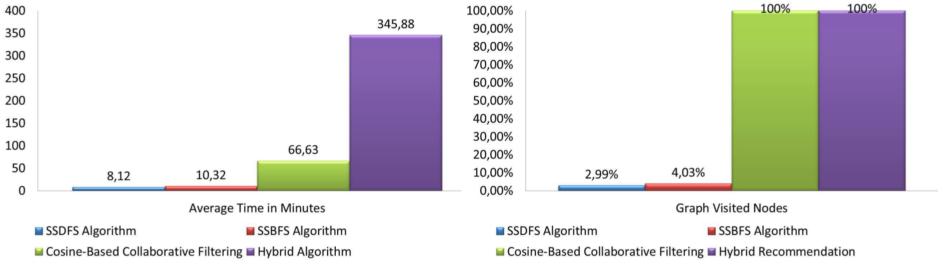


Fig. 7. Comparison of runtime and visited nodes between SSDFS, SSBFS, CF and HR

then it is likely to be high for the nodes ‘close’ to it. Since, in a breadth first strategy, these nodes are visited earlier than in a depth first strategy, the output of SSBFS contains more true positives can that of SSDFS.

5 Related Work

In 1999, IRA (Intelligent Recommendation Algorithm, [10]) was proposed as a graph-based collaborative filtering recommender system, in which a breadth-first search algorithm is used to compute the shortest paths between nodes. User-item bipartite graph and one-mode projection are used in a movie recommender system proposed in [11]. In this system a recommendation graph has been defined as the sum of the bipartite graph and the one-mode projection graph, then the shortest path algorithm has been applied on this recommendation graph. In [1], a random walk algorithm is proposed to recommend items in a trust network. This algorithm recommends items based on ratings expressed by trusted friends, using random walk and probabilistic item selection.

Other recommender systems include semantic aspects, in addition to collaborative filtering aspects. In [12] a recommendation algorithm is introduced for collaborative URL tagging. In this system, user interests are modeled according to their social ties and the vocabularies they use to tag URLs. In [13] similar tags are grouped in clusters, these tag clusters are used as intermediate sets between users and items. In [14] the authors propose to represent the users by a vector of scores assigned to topics taken from domain ontology; then a semantic similarity measure is used in a semantic-based recommender system.

We also refer to [15] for a survey on recommender systems where three main categories of recommender systems (namely content-based, collaborative-filtering or social, and hybrid) are introduced.

6 Conclusion

In this paper we have proposed two semantic social recommendation algorithms called *Semantic Social Depth First Search* and *Semantic Social Breadth First Search*. These algorithms work on a social network, assuming that users and items are described via an ontology. We applied these algorithms on a real dataset, and our experiment results show that our algorithms give good F-Measure values, while outperforming collaborative filtering and hybrid recommendation algorithms.

In our future work, we intend to consider other social network analysis measures and other datasets, so as to better assess our approach. Moreover, the study and test of other heuristics for graph exploration is another important issue that will be investigated in the near future.

References

1. Jamali, M., Ester, M.: Trustwalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 397–406. ACM, NY (2009)
2. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User Profiles for Personalized Information Access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 54–89. Springer, Heidelberg (2007)
3. Zuber, V.S., Faltings, B.: OSS: A Semantic Similarity Function based on Hierarchical Ontologies. In: Proceedings of IJCAI 2007, pp. 551–556 (2007)
4. Jiang, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: International Conference Research on Computational Linguistics (ROCLING X) (1997)
5. Ramasco, J.J.: Social inertia and diversity in collaboration networks. The European Physical Journal Special Topics 143, 47–50 (2007)
6. Newman, M.E.J.: Networks An Introduction. Oxford University Press (2010)
7. Cano, P., Celma, O., Koppenberger, M., Buldú, M.J.: Topology of music recommendation networks. Chaos An Interdisciplinary Journal of Nonlinear Science 16 (2006)
8. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004)
9. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295. ACM, New York (2001)

10. Aggarwal, C.C., Wolf, J.L., Wu, K., Yu, P.S.: Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering. In: KDD 1999: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 201–212. ACM, San Diego (1999)
11. Mirza, B.J., Keller, B.J., Ramakrishnan, N.: Studying Recommendation Algorithms by Graph Analysis. Journal of Intelligent Information Systems 20(2), 131–160 (2003)
12. Stoyanovich, J., Yahia, S.A., Marlow, C., Yu, C.: A study of the benefit of leveraging tagging behavior to model users'Interests in del.icio.us. In: AAAI Spring Symposium on Social Information Processing (2008)
13. Shepitsen, A., Gemmell, J., Mobasher, B., Burke, R.: Personalized recommendation in social tagging systems using hierarchical clustering. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, pp. 259–266. ACM, New York (2008)
14. Ziegler, C.N., Lausen, G., Lars, S.T.: Taxonomy-driven computation of product recommendations. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM 2004, pp. 406–415. ACM, New York (2004)
15. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)

A Methodological Framework for Statistical Analysis of Social Text Streams

Sophia Kleisarchaki[1,2], Dimitris Kotzinos[1,3],
Ioannis Tsamardinos[1,2], and Vassilis Christophides[1,2]

[1] Institute of Computer Science, FORTH
{kleisar,kotzino,tsamard,christop}@ics.forth.gr
[2] Computer Science Department, University of Crete, Greece
[3] Department of Geoinformatics and Surveying, TEI Serres, Greece

Abstract. Social media are one of the main contributors of user generated content; providing vast amounts of data in daily basis, covering a wide range of topics, interests and events. In order to identify and link meaningful and relevant information, clustering algorithms have been used to partition the user generated content. We have identified though that these algorithms exhibit various shortcomings when they have to deal with social media textual information, which is dynamic and streaming in nature. Thus we explore the idea to estimate the algorithms' parameters based on observations on the clusters' properties' (like the centroid, shape and density) evolution. By experimenting with the clusters' properties, we propose a methodological framework that detects the evolution of the clusters' centroid, shape and density and explores their role in parameters' estimation.

Keywords: twitter, clustering algorithm, centroid, shape, density.

1 Introduction

We are witnessing an unprecedented growth of interest in social media[1] enabling people to achieve a *near real-time information awareness*. Several online networking sites (e.g. Facebook), micro-blogging applications (e.g. Twitter) and Social news (e.g. Digg) produce on a daily basis vast amounts of user-generated textual content. Identifying *topics* of conversation in social text streams and monitoring how they evolve over time have attracted both scientific and industrial interest.

Twitter enables users to post short textual messages (up to 140 characters), known as *tweets*, to update their followers with their findings, thinking and comments on some topics. Topics cover in general, a wide variety of *real-world events* [3] ranging from popular, widely known events (e.g., related to worldwide or national breaking news, sports or music events) to happenings that might receive no coverage in traditional news outlets (e.g., a local social gathering, an annual convention, or a community-specific reunion). According to a recent

[1] en.wikipedia.org/wiki/Socialmedia

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 101–110, 2013.

empirical study the majority of Twitter users post messages regarding their personal concerns and matters, whereas only a smaller percentage actually share information of general interest [17]. In this vibrant information space, topics are usually modelled as sets (or sequences) of words whose co-occurrence distribution in tweets become bursty [16,7,12] or as clusters of tweets exhibiting an important similarity on their textual content by considering the user network structure [1,23,21] as well as their temporal and spatial metadata of tweets [8,2].

Previous related work have mostly focused on the scalability of algorithms for clustering large volumes of tweets[2] either online [3] based on a priori knowledge of the topics of interest [20,19] or offline [18] by discovering a variable number of topics. Less attention has been given to the actual effectiveness of the clustering algorithms as the conversation among a volatile community of users evolves in real social streams. This can be partially attributed to the lack of gold standards for social messages (such as TDT[3] or TREC[4] benchmarks). In this paper we are studying the behaviour of two prototypical clustering algorithms using a real workbench collected during a 9 months campaign in 2010-2011 via Twitter public streams API[5]. Our workbench comprises 9.056.914 English tweets on major physical (earthquake in Japan, 7.37%), political (Libya revolution, 2.6%) and athletic (Champions League, 0.03%) events. Then, using the Alias [9] method we extract specific samples of our workbench that simulates various live-stream scenarios regarding the number of the Twitter #tags and arrival rate of tweets that need to be analysed as well as their temporal overlap as they were delivered to the users. We are particularly interested in understanding how the *number* and the *entropy* of the detected clusters is affected by changes in their *centroid*, *shape* and *density* as the *granularity* and dynamics of conversations in tweets (i.e., sub-topics under the same #tag) evolve over time. Addressing temporal aspects of social text streams is a pre-requisite for much-needed models of conflicting and consensual information, as well as for modelling change in user interests and is a relatively under-researched problem [5].

1.1 Problem Statement

We are focusing on the clustering of tweets based on their textual content since it is the most informative part of social messages and the #tags under which they are posted to index them in high-level thematic categories. Compared to these categories, the detected clusters are expected to capture more fine-grained topics (i.e. sub-topics) of the conversation conducted in a social stream. To this end, we use two state-of-the-art clustering algorithms, namely k-means [11] and TStream [25], over a sample of 10000 tweets from our workbench which cover three thematically independent #tags occurring simultaneously in time with much different arrival rates and volumes. These algorithms aim to partition

[2] goo.gl/BkNEq

[3] goo.gl/Oq6Bk

[4] trec.nist.gov

[5] goo.gl/D4bMp

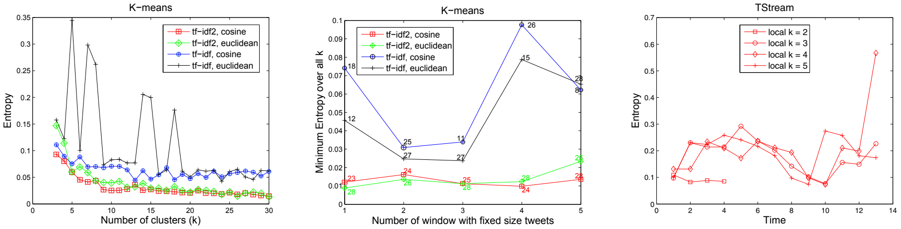
incoming tweets to a fixed number k of (sub-)topics. As we will see in the sequel the best possible k w.r.t. the resulting clustering quality strongly depends on the calibration parameters of the two algorithms for our sample dataset. As quality criterion we have chosen the conditional entropy [24] of the #tag of a tweet given its cluster $H(Tag \mid C)$ with the following reasoning: a low $H(Tag \mid C)$ implies that knowing the cluster of a tweet, there is little uncertainty about its #tag. Thus, this metric does not penalize for a clustering that partition the tweets published under the same #tag to several clusters, corresponding to multiple sub-topics, but penalizes only when tweets from different #tags are placed in the same cluster.

K-means is based on an iterative partitioning of the input points into k clusters, in the aim to minimize the sum, over all clusters, of the within-cluster sums of *point to cluster centroid distances.* To estimate the number of clusters resulting to a minimal entropy, we evaluated the algorithm for several k-values carrying out 20 iterations over our sample dataset. Since the within-cluster distance is affected both by the employed weighting scheme (WS) of terms and the similarity metric, in Figure 1(a) we report several alternative choices, among which $tf \cdot idf_2$[6] and cosine similarity [15] exhibit for our dataset the best possible clustering quality (i.e. the lowest entropy). K-means fails to detect clusters for the #tag with the smaller number of tweets that it just constitutes the 0.03% of the total dataset. It worth mentioning that in this experiment the entropy does not monotonically drop as the number of clusters increases. That is due to the way entropy is computed: it is the estimated rather than the true entropy since the true probabilities of the population distribution are unknown. Furthermore, k-means is a heuristic algorithm which does not guarantee an optimal clustering. As a result in the estimated entropy calculations a variance is introduced that increases with k, causing statistical fluctuations. K-means quality is substantially affected by the number of dimensions of the Feature Space (FS) employed to compute the distance of points. The entropy reported in Figure 1(a) has been computed for a constant FS with the 500 most frequent occurring terms in the entire sample dataset after ignoring stopwords. Since the vocabulary of tweets actually evolves over time along with the drift in user conversations, we need to dynamically adjust not only the number of detected clusters but also the FS on which they are computed.

To this end, Figure 1(b) depicts the entropy (y-axis) when k-means is independently executed on consecutive, non-overlapping time-windows with 2000 tweets each (x-axis) by exploring an adjustable FS with the most frequent 500 terms occurring in each window[7]. For all possible combinations of terms'

[6] $tf \cdot idf_2$ differentiates from the well-known $tf \cdot idf$ weighting scheme only in the first factor tf that refers to the word's frequency occurrence in the entire corpus instead of the single tweet. Some related works [22] use $df \cdot idf$ WS instead, where df counts the number of tweets containing a word. However multiple appearances of the same word in a tweet are rare making the two WS similar.

[7] We observed that increasing the dimensions over 500 has no significant benefit to clustering, while fewer than 500 dimensions result to too many zero weighting vectors.



(a) K-means results for static analysis (constant FS) of different WS: $tf \cdot idf_2$ & cosine robust in time

(b) K-means results for dynamic analysis (adjustable FS) of different WS: $tf \cdot idf_2$ & cosine robust in time

(c) Clusterings of TStream in different windows are related: calibration of parameters is a critical task

Fig. 1. Entropy for k-means & TStream algorithms

weighting schemes and similarity metrics we report above the four curves the number of clusters for which the smallest estimated entropy is observed in each window. The fluctuation of entropy for the two $tf \cdot idf$ curves is more important than for $tf \cdot idf_2$ while the absolute entropy values computed in the last window of 2000 tweets are very close to (vs. considerably diverge from) the values of the latter (vs. former) computed in the entire dataset of 10000 tweets (see Figure 1(a)). Note that in windows 3, 4 $tf \cdot idf_2$ curves seem to be also resistant to bursty arrival of tweets (from #tag Japan). It should be however stressed that k-means produces un-nested and non-overlapping spherical clusters without any information regarding the relationship among them. In addition the independent execution of k-means in each window completely forgets past tweets in previous ones making difficult to understand the evolving shape and density of the detected clusters across windows.

For this reason we have additionally considered the hierarchical clustering algorithm TStream, which detects spherical clusters of high-similarity (i.e. subtopics) nested within wider ones (i.e. topics). In order to update such a two-level clusters hierarchy when a number of novel data is detected, TStream periodically re-organizes either the first or the second level of clusters by recomputing their FS based on the new collection and the memory of the W latest data ($tf \cdot idf_2$ weighting [8]). Table 1 presents a high number of parameters that need to be initialized in advance for the TStream algorithm. A first effort towards the parameters calibration has been made based on the analysis of Section 2 and the knowledge of k-means results. The *global* and *local container* (GC/LC) has been set to 600 since this number of novel tweets has been proven statistically significant in our dataset (see Section 2) to trigger a first or second level re-clustering (i.e. topic evolution). The GC cosine similarity threshold was set to the average similarity of k-means clusters (cosine, $tf \cdot idf_2$) containing tweets with the same #tag for which the lowest entropy ($k = 30$) is exhibited while the LC similarity threshold was defined as the minimum cosine metric detected in the clusters. The number of the latest tweets (W) considered in case of re-clustering was empirically set to the size of two windows. The *kglobal* parameter refers to

[8] For consistency reasons, we changed the $tf \cdot idf$ WS used by TStream into $tf \cdot idf_2$.

the number of the first-level clusters capturing distinct #tags, whereas *localk* defines the number of second-level clusters where at least two sub-topics were detected by k-means for each #tag. We used 2000 tweets as the initialization step (*initialDocNo*) of the algorithm and tumbling windows (Wsz) of the same size. The dimensions of clusters' centroid (*centroidSz*) as well as the size of the input word vectors (*WordSz*) were set to 500.

Table 1. Parameters of TStream algorithm

global σ (GC)	**local σ (LC)**	**global δ (GSim)**	**local δ (LSim)**	**W**	**WSz**
600	600	0.52	0.7	2	2000
global k	**local k**	**initialDocNo**	**centroidSz**	**WordSz**	
2	$[2, 5]$	2000	500	500	

Figure 1(c) depicts the entropy for the top-level clusters detected by TStream for various values of *localk*. The entropy of TStream clustering appear to be in most cases at least one order of magnitude higher than k-means. Unlike Figure 1(b), where the x-axis refers to sequential fix-sized windows, in Figure 1(c) the x-axis refers to the relative time points in which different volumes of tweets are clustered. Thus, the various curves of the entropy are not directly comparable. We can observe downward or upward trends in the entropy caused by the global, local or no re-clustering decisions of TStream depending on the actual parameters' value and the tweets of each time window. For instance, for *localk*=2 no re-clusterings are performed as there are not enough *novel* tweets above the novelty thresholds (GC/LC) to trigger such process. Since neither the global nor the local hierarchy is re-organized, the new incoming tweets are clustered into the already existing clusters and thus the curve (*localk*=2) shows the entropy at the time points where a new window ($Wsz = 2000$) is added. We can observe that this is the only case where TStream results to the same clustering quality as k-means for k=4. In particular, TStream maintains two top-level clusters corresponding to the high-level topics 'Japan' and 'Libya' each one containing second-level clusters with tweets sharing the same #tag, while k-means results to three clusters containing tweets from the #tag 'Japan' and one from 'Libya' with only few miss-classifications (<300 tweets).

In both algorithms, to improve the clustering quality a thorough calibration of their input parameters is needed which implies to understand how core cluster properties evolve over time. As we have seen a *static* parameters calibration is not always able to improve the clustering quality. For this reason, we propose an original methodological framework for recognizing how clusters *centroid*, *shape* and *density* evolve along with the *granularity* and *dynamics* of user conversations in real tweets. We believe that this framework is essential in order to enable a *dynamic* adaptation of the parameters impacting the quality of the clustering results achieved by different algorithms.

The rest of the paper is organized as follows. Section 2 presents our experimental findings regarding core evolution aspects of clusters that could be detected in time and count-based windows of sample datasets. Section 3 concludes the paper and presents some pointers for future research in the area.

2 Analysis of Clusters' Core Aspects

In an effort to better estimate the parameters of the algorithms, we study the core aspects of the clusters emphasizing on the investigation of *cluster's centroid*, *shape* and *density evolution* over time in order to understand which of them and to what extent can be used to experimentally estimate those parameters. Semantically, the centroid of a cluster summarizes the discussion of a topic by providing the vocabulary consisting of the most representative words. The various opinions of the users as well as their opinions' convergence or discrepancy over time are illustrated in the cluster's *shape* and *density*. Intuitively we expect that the textual social content, being of dynamic nature, results in an evolving vocabulary and discussion of varying flavours. Hence, the clusters are shifting in space over time with shape shrinking and expanding dynamically. Thus, we firstly assume that the centroid and shape of the clusters are constant (*null hypothesis* H_0) and then we try to reject H_0 by applying proper statistical hypothesis tests.

2.1 Centroid Evolution

The centroid is a representative point that summarizes the contents of the cluster and defines a geometric center of the cluster's shape. It is not necessarily member of the dataset and the most common technique to define its value is by calculating the mean of the cluster's data points. Therefore, the study of centroid trajectory semantics reveals the evolution of the representative points and thus the evolution of the topic's vocabulary. A non significant movement of the centroid indicates a static topic summary of an already shaped public opinion where the most representative words are repeated in the same volume of posts and are of the same weighting importance. On the other hand, a shift of centroid shows a differentiation in discussion over time with some words increasing and other words loosing their significance. Thus, the vocabulary of the users changes as the time progresses forcing the centroid to move in its multidimensional space. Centroid evolution is an interesting property of the cluster and a critical parameter of the clustering algorithms.

Assume that a data stream consists of a set of multidimensional points $p_1, ...,$ $p_i, ...$ arriving at time stamps $t_1, ..., t_i, ...$ and $p_i = (p^i_1 ... p^i_d)$. Our alternative hypothesis (H_1) states that by updating the centroid with the new incoming points its position changes dynamically in space and time in a way that it is unlikely to have occurred by chance. In order to accept H_1 we test the null hypothesis H_0 by applying the permutation-based two-samples Hotelling's T-squared [13] statistical test over the distributions of the succeeding Sliding Window (SW) models. Furthermore, we study the occurrence of centroid's evolution utilizing the bursT [14] dynamic weighting scheme.

In the **count-based sliding window** model, only the most recent N tweets are considered at any time, where N is the size of the window. We introduce a slip step where n_{step} number of new tweets are added to the window, while the oldest n_{step} are dropped out and no longer contribute to the clustering.

In the **time-based window** model, the window M_i starts at time t_i and ends at time t_j so that the time interval $t_j - t_i$ of the tweets inside the window corresponds to a constant value Δt at any time. We introduce a slip step t_{step} as a time interval during which new points are added to the window and the oldest are removed, so that the time interval of the window remains constant.

For the first count-based SW, $N = 2000$ tweets are considered derived from the highest volume of #tag (i.e. Japan) at any time for various values of the slip step n_{step}. For the latter, a constant $\Delta t = 1day$ of the same #tag is selected and the t_{step} spans from 1h to 1day. The value of the time interval Δt is an intuitive trade-off between the time period of the collection and its arrival rate. The first SW is utilized to determine the top 500 terms of the vocabulary. Inside the rest SWs the weights of the top 500 words ($tf \cdot idf_2$) are recomputed and 1000 permutations are performed. Table 2 demonstrates the range of the p-values (p_0) extracted from the execution of the statistical test for all the windows of the sample of each sliding model. The p_0 is the probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. We can reject the null hypothesis when the p-value is less than the significance level α set to 0.05. We note that for $n_{step} \geq 600$ the p_0 is below α and thus we can safely reject the null hypothesis. The windows have statistically significant different centroids for $n_{step} \geq 600$ corresponding to data of $t_{step} = 24h$. The SW models reveal that there is a minimum number (threshold) of tweets that bring the evolution of the centroid either in terms of an absolute number or of time. Furthermore, the use of count-based SW model allows the processing of the data in batches giving memory guarantees, whereas the time-based windows give the advantage of tuning the data freshness and study the centroid's behaviour of varying size windows.

Table 2. P-values of the permutation-based Hotelling's test

n_{step}	Count-based (p_0 range)	t_{step}	Time-based (p_0 range)	bursT
200	[0, 1]	1h	1	[0.051, 1]
300	[0, 1]	12h	[0, 0.995]	[0, 0.997]
400	[0, 0.994]	24h	0	[0, 0.179]
500	[0, 0.356]			
600	0			

Conceptually, the statistically significant difference of the centroids means that either the weights of the words or the distribution of the tweets change. Some words might now be appearing as more significant in the discussion, while others less because of the sparse occurrences. The contribution of the terms in the discussion changes over time. Thus, the cluster's centroid of pre-defined tags evolves in a multi-dimensional space over time in a statistically significant way.

Instead, the dynamic weighting method partially absorbs the evolution of the centroid by penalizing the words that are no more active and increasing those that are currently in use. This behaviour causes the centroid to re-adjust its position in a non statistically significant way (see Table 2). Nevertheless, there are several sliding windows for $t_{step} \geq 12h$ with centroids that differ significantly, indicating that the dynamic weighting scheme can not fully compensate the changes and keep the centroid stagnant in space over time.

2.2 Shape Evolution

The shape of the cluster gives a notion of points distance from the centroid. An almost constant shape of non significant evolution reveals a cluster where people discuss about a topic with the same interest over time using the same representative vocabulary. On the other hand, an evolving shape indicates a discrepancy between the discussed topic in the past and in the present. An expansion of shape means that the points are getting away from the centroid and simultaneously increase their distance, meaning that far from the average opinions are evolving and discordant views exist among users. A shrinkage of the cluster's shape shows that the users tend to agree with each other and are shrunk around the same vocabulary with a representative opinion expressing the majority of them.

In order to test our initial assumption of shape stagnation we perform the two sample Kolmorgorov-Smirnov test [9], which compares two different distributions under the null hypothesis that they arrive from the same continuous distribution. Table 3 summarizes the range of the extracted p_0 for each SW model. We observe that the clusters inside the windows evolve in shape with the addition of more than 400 new points or equivalent with data of one day. It is worth mentioning that the evolution of shape happens earlier in time than the movement of centroid, indicating that the cluster first changes its shape and then moves its centroid. On the contrast, no modification of shape is observed using the dynamic weighting as it encapsulates the evolution by re-positioning the points of the cluster.

Table 3. P-values of the Kolmogorov-Smirnov test

n_{step}	Count-based (p_0 range)	t_{step}	Time-based (p_0 range)	bursT
200	$[4.48 \cdot 10^{-244}, 0.98]$	1h	$[8.89 \cdot 10^{-14}, 1]$	$[1.31{\cdot}10^{-3}, 0.99]$
300	$[1.11 \cdot 10^{-268}, 0.57]$	12h	$[1.04 \cdot 10^{-20}, 0.23]$	$[8.35{\cdot}10^{-6}, 0.12]$
400	$[0, 0.027]$	24h	$[9.68 \cdot 10^{-25}, 0.003]$	$[1.39{\cdot}10^{-3}, 0.53]$
500	$[2.08 \cdot 10^{-285}, 0.0014]$			
600	$[5.8 \cdot 10^{-312}, 5.3 \cdot 10^{-5}]$			

2.3 Density Evolution

So far, the study of core evolution aspects of clusters inside the SWs revealed a shift of centroid and shape. Subsequently, we study how the cluster's density influences its shape tendency to shrink or expand over time and space.

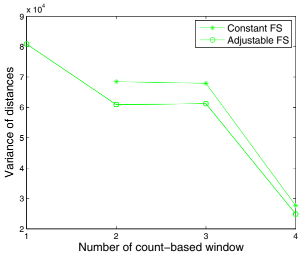


Fig. 2. Variance of distances from cluster's centroid

Figure 2 shows the variation of distances from centroid for the different count-based SWs ($\Delta t = n_{step} = 2000$) of one #tag utilizing a constant FS. The first time window is used for the initialization step of the vocabulary. We note that the clusters shrink over time reducing their volume. In order to ensure that the shrinkage of the clusters' shape is not due to the lack of representative terms

[9] en.wikipedia.org/wiki/Kolmogorov-Smirnov_test

we plot the variation of distances for an adjustable FS. Therefore, we can state that users tend to agree with each other as time progresses.

Furthermore, we notice that even with an adaptable FS the reduction of distances' variance happens. In fact, the curve of the distance variance for the refined FS is always below the curve of the constant FS causing more compact clusters.

3 Conclusions and Future Work

In this paper we proposed a methodological framework for studying the evolution aspects of the clusters' centroid, shape and density properties. We showed that the clusters' centroid and shape evolve over time in a statistically significant way inside a multi-dimensional space where topic shifts and drifts are exhibited. We noticed that the dynamic weighting scheme of bursT partially absorbs the evolution for several time windows by re-adjusting the position of the weighted tweets. Furthermore, we observed that clusters tend to shrink as time progresses even with the use of an adjustable feature space indicating a common vocabulary among users. Moreover we tried to identify which features of the algorithms (i.e. similarity and novelty thresholds, number of clusters etc) are affected by the evolution of the social textual stream and understand their shortcomings in order to be able to suggest better parameters' tuning; providing a more complete methodological framework for such analysis.

We plan to continue our work by also evaluating non partitioning clustering algorithms. For instance, the topic extraction task in textual corpus is addressed through probabilistic models where each document is associated with a probability distribution over topics. Latent Dirichlet Allocation [4] is the most well known topic model and a preliminary study has been made regarding its applicability on social text streams [19]. Furthermore, we plan to study algorithms that dynamically adjust the clusters' centroid [6] by implicitly assuming that its distribution change over time. Last but not least, we are interested in studying twitter-specific similarity measures [10], where the content proximity of tweets is defined as a combination of textual (i.e. tweet) and non textual (i.e. network structure, spatio-temporal) information.

Acknowledgments. We acknowledge the contribution of Prof. M. Spiliopoulou and Mr. M. Zimmermann for providing us with the implementation of TStream algorithm. This work was partially supported by the INGEOCLOUDS project (Project reference: 297300) under the CIP-ICT-PSP.2011.4.1

References

1. Aggarwal, C., Subbian, K.: Event detection in social streams. In: Proc. of SDM (2012)
2. Becker, H., Naaman, M., Gravano, L.: Learning similarity metrics for event identification in social media. In: Proc. of the 3rd WSDM, pp. 291–300. ACM, NY (2010)
3. Becker, H., Naaman, M., Gravano, L.: Beyond trending topics: Real-world event identification on twitter. In: 5th ICWSM. AAAI (2011)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)

5. Bontcheva, K., Rout, D.: Making sense of social media streams through semantics: a survey. Semantic Web Journal (2012)
6. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: SIAM Conference on Data Mining, pp. 328–339 (2006)
7. Cataldi, M., Di Caro, L., Schifanella, C.: Emerging topic detection on twitter based on temporal and social terms evaluation. In: Proc. of the 10th MDMKDD, pp. 4:1–4:10. ACM, NY (2010)
8. Chen, L., Roy, A.: Event detection from flickr data through wavelet-based spatial analysis. In: Proc. of the 18th ACM CIKM, pp. 523–532. ACM, NY (2009)
9. Devroye, L.: Sample-based non-uniform random variate generation. In: Proc. of the 18th WSC, pp. 260–265. ACM, NY (1986)
10. Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.-Y.: An empirical study on learning to rank of tweets. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010, pp. 295–303. Association for Computational Linguistics, Stroudsburg (2010)
11. Hartigan, J.A., Wong, M.A.: A K-means clustering algorithm. Applied Statistics 28, 100–108 (1979)
12. He, Q., Chang, K., Peng Lim, E., Zhang, J.: Bursty feature representation for clustering text streams
13. Hotelling, H.: The Generalization of Student's Ratio, pp. 360–378 (August 1931)
14. Lee, C.-H., Wu, C.-H., Chien, T.-F.: BursT: A dynamic term weighting scheme for mining microblogging messages. In: Liu, D., Zhang, H., Polycarpou, M., Alippi, C., He, H. (eds.) ISNN 2011, Part III. LNCS, vol. 6677, pp. 548–557. Springer, Heidelberg (2011)
15. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
16. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream. In: Proc. of SIGMOD, pp. 1155–1158. ACM, NY (2010)
17. Naaman, M., Boase, J., Lai, C.-H.: Is it really about me?: message content in social awareness streams. In: Proc. of CSCW, NY, USA, pp. 189–192 (2010)
18. Petrović, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: HLT: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 181–189 (2010)
19. Ramage, D., Dumais, S., Liebling, D.: Characterizing microblogs with topic models. In: ICWSM (2010)
20. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proc. of the 19th WWW, pp. 851–860. ACM, NY (2010)
21. Sayyadi, H., Hurst, M., Maykov, A.: Event detection and tracking in social streams. In: Proc. of the ICWSM. AAAI (2009)
22. Weng, J., Yao, Y., Leonardi, E., Lee, F., Lee, B.-S.: Event detection in twitter. Development (98), 401–408 (2011)
23. Zhao, Q., Mitra, P., Chen, B.: Temporal and information flow based event detection from social text streams. In: Proc. of the 22nd AAAI, vol. 2, pp. 1501–1506. AAAI Press (2007)
24. Zhao, Y., Karypis, G.: Criterion functions for document clustering: Experiments and analysis. Technical report (2002)
25. Zimmermann, M., Ntoutsi, I., Siddiqui, Z.F., Spiliopoulou, M., Kriegel, H.-P.: Discovering global and local bursts in a stream of news. In: Proc. of the 27th SAC, pp. 807–812. ACM, NY (2012)

How to Extract Relevant Knowledge from Tweets?

Flavien Bouillot[1], Phan Nhat Hai[1,4], Nicolas Béchet[2], Sandra Bringay[1,3],
Dino Ienco[1,4], Stan Matwin[5], Pascal Poncelet[1],
Mathieu Roche[1], and Maguelonne Teisseire[1,4]

[1] LIRMM – CNRS, 161 rue Ada, Montpellier, France
{bringay,bouillot,poncelet,mroche}@lirmm.fr
[2] Univ. Caen Basse-Normandie, Caen, France
nicolas.bechet@unicaen.fr
[3] Univ. Montpellier 3, Montpellier, France
[4] IRSTEA - UMR TETIS, Montpellier France
{nhat-hai.phan,dino.ienco,maguelonne.teisseire}@teledetection.fr
[5] University of Ottawa, Ontario, Canada
stan@site.uottawa.ca

Abstract. Tweets exchanged over the Internet are an important source of information even if their characteristics make them difficult to analyze (e.g., a maximum of 140 characters; noisy data). In this paper, we investigate two different problems. The first one is related to the extraction of representative terms from a set of tweets. More precisely we address the following question: *are traditional information retrieval measures appropriate when dealing with tweets?*. The second problem is related to the evolution of tweets over time for a set of users. With the development of data mining approaches, lots of very efficient methods have been defined to extract patterns hidden in the huge amount of data available. More recently new spatio-temporal data mining approaches have specifically been defined for dealing with the huge amount of moving object data that can be obtained from the improvement in positioning technology. Due to particularity of tweets, the second question we investigate is the following: *are spatio-temporal mining algorithms appropriate for better understanding the behavior of communities over time?* These two problems are illustrated through real applications concerning both health and political tweets.

1 Introduction

In recent years, the development of social and collaborative Web 2.0 underlines the central and active role of users in collaborative networks. Blogs to spread diaries, RSS news to track last information on a specific topic, tweets to publish social actions, are now extremely widespread. Easy to create and manage these tools are used by Internet users, businesses or other organizations to communicate about themselves and the growing use of this technology starts to influence

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 111–120, 2013.

many aspect of the real life. Furthermore, this data represents an important source of information that can be exploited in the decision making process.

Since its introduction in 2006, the Twitter website[1] has become so popular that it is currently ranked as the 10^{th} most visited site over the world[2]. In January 2012, Twitter has been visited 2.5 billion times and in October 2011, more than 250 million tweets are posted every day with a user base of about 300 million people. Basically, Twitter is a platform for microblogging. It means that it is a system for sharing information where users can either follow other users who post short messages (i.e. 140 characters), or can be followed. When a user follows a person, the user receives all messages from this person, and in turn, when that user tweets, all his followers will receive the messages. Tweets are associated with meta-information that cannot be included in messages (e.g., date, location, etc.) or included in the message in the form of tags having a special meaning. For example the tag *@username* means that you are sending a message to a particular user, the *# topic* assigns a specific topic, *RT* means that the message was re-tweeted, i.e. send to all the followers.

Actually, by taking into account all this meta-information, we can observe that tweets can be represented in a multidimensional way with, for instance, one dimension for the location, one dimension for the time, one dimension for the set of words used, etc. In this context, different systems were proposed to analyze this flow of information [1,2,3]. For instance, they can focus on event detection [3], Name Entity recognition [4], can combine different types of information such as timeline and sentiment features [5] or even analyze propagation from specific features such as hashtags [6].

Nevertheless all these approaches do not really exploit their multidimensional characteristics. In [7], we propose to address these multidimensional characteristics by focusing on datawarehouses [8] since they provide very efficient tool for the storage and analysis of multidimensional and historized data. Our main goal was to using the facilities provided by these tool to manipulate a set of indicators (measures) according to the different dimensions obtained from tweets and for which we can be provide some hierarchies. Furthermore associated operators (e.g., Roll-up, Drill-down, etc.) allow an intuitive navigation on different levels of the hierarchy. In this paper we focus on the problems associated with the definition of such measures when dealing with hierarchies and propose some extension that are well adapted to our concern. In order to illustrate how such a tool can be useful we report some results on experiments conducted on the health domain.

By using a multidimensional tool we are able to really help the end user to analyze the data over time. Nevertheless one problem remains. As we are provided with a huge amount of data, applying data mining techniques seems relevant to highlight knowledge hidden in the data. Among the different techniques we investigate if pattern mining approaches are appropriate to understand users' behaviors. More precisely we focus on trajectory mining algorithms to analyze

[1] `http://twitter.com`

[2] `http://www.alexa.com/topsites`

behavior of communities of user. Basically they are defined for dealing with spatio-temporal data. We show that measures defined for tweet datawarehouse can also be used to define new kinds of trajectories related to tweet terms rather than spatial information. We illustrate some trajectories that can be extracted from the analysis of the evolution of French political communities[3] over Twitter during 2012 which was particularly important for French political communities dues the two main elections: Presidential and Legislative.

The remainder of this paper is organized as follows. Section 2 investigates how multidimensional characteristics can be handled and focus on different kinds of useful measures. It also illustrates some results by using tweets related to diseases. We address the problem of pattern mining in Section 3. After a brief presentation of trajectory approaches we illustrate how these patterns can be useful to understand users' behaviors over time. Finally Section 4 concludes the paper and presents future work.

2 Towards a Datawarehouse for Analyzing Tweets

In this section we propose a minimal model for dealing with multidimensional characteristics of tweets. We mainly focus on three dimensions: the dimension Word corresponding to the set of words (or terms) that can be extracted from tweets, the dimension location that can be extracted from metadata and the time dimension. Furthermore we illustrate the model through tweets expressed in the health domain.

2.1 The Model

According to [9], a fact table F is defined on the schema $D = \{T_, \ldots, T_n, M\}$ where T_i $(i = 1, .., n)$ are the dimensions and M stands for a measure. Each dimension T_i is defined over a domain D partitioned into a set of categories C_j. We thus have $D = \cup_j C_j$. D is also provided with a partial order $\sqsubseteq_D$ to compare the values of the domain D. Each category represents the values associated with a level of granularity. We note $e \in D$ to specify that e is a value of the dimension D if there is a category $C_j \subseteq D$ such that $e \in \cup_j C_j$. Note that two special categories are distinguished and are present on all dimensions: $\perp_D$ et $\top_D \in C_D$ corresponding respectively to the level of finer and higher granularity. In our approach, the partial order defined on the domains of the dimensions stands for the inclusion of keywords associated to the values of the dimensions. Thus, let $e_1, e_2 \in \cup_j C_j$ be two values, we have $e_1 \sqsubseteq_D e_2$ if e_1 is logically contained in e_2.

Example 1. *Figure 1 illustrates the location dimension having a hierarchy such that* $\perp_{location} = City \leq State \leq Country \leq \top_{location}$. *The values of the dimension are* $dom(Localisation) = \{$*New York, Albany, Los Angeles, Northeastern,*

[3] This work is a part of the PoloP Project (*Political Opinion Mining*) which aims to cope with the analysis of the evolution of French political communities over Twitter during 2012 both in terms of relevant terms, opinions, behaviors.

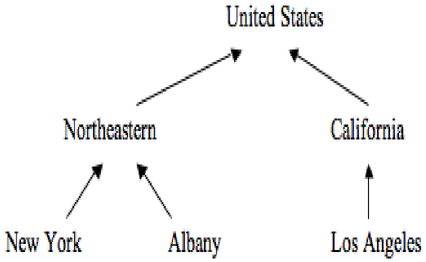


Fig. 1. A part of the hierarchy associated to the location dimension

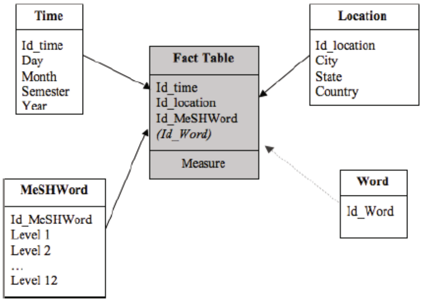


Fig. 2. An example of a schema for dealing with tweets in the health domain

California, United States, ...} divided into these categories (levels of granularity) as follows: City = $\{NewYork$, *Albany*, *Los Angeles*$\}$, *State*=$\{$*Northeastern, California*$\}$, *Country* = $\{$*United States...*$\}$. *The partial order* $\sqsubseteq_D$ *over the values of dimensions can be generalized to categories: for* $C_1, C_2 \in C_D$, *we thus have* $C_1 \leq_D C_2$ *if* $\exists e_1 \in C_1, e_2 \in C_2$ *such as* $e_1 \sqsubseteq_D e_2$. *For example, we have Los Angeles* $\sqsubseteq_D$ *California* $\sqsubseteq_D$ *United States* $\sqsubseteq_D \top$. *The taking into account of the dynamic hierarchy is such that all categories of this dimension must respect the defined partial order.*

Figure 2 illustrates the associated schema. We find the dimension *location* and the dimension *time* as $\perp_{temps} = day \leq month \leq semester \leq year \leq \top_{temps}$. For the hierarchy, we use the MeSH (Medical Subject Headings)[4] National Library of Medicine's controlled vocabulary thesaurus. It consists of sets of terms naming descriptors in a twelve-level hierarchy that permits searching at various levels of specificity. At the most general level of the hierarchical structure are very broad headings such as "Anatomy" or "Mental Disorders". More specific headings are found at more narrow levels, such as "Ankle" and "Conduct Disorder". In 2011, 26,142 descriptors are available in MeSH. There are also over 177,000 entry terms that assist in finding the most appropriate MeSH Heading, for example, "Vitamin C" is an entry term to "Ascorbic Acid".

2.2 Some Proposed Measures

Traditionally, the *TF-IDF* measure (Term Frequency - Inverse Document Frequency), introduced by [10], is a very useful measure that giving greater weight to the discriminant terms and can thus be well adapted to our concern. As a first step, it is necessary to compute the frequency of a term (*Term Frequency*) corresponding to the number of occurrences of the term in the document[5].

[4] `http://www.nlm.nih.gov/pubs/factsheets/mesh.html`

[5] Here *document* is used to be compliant with the original definition of the *TF-IDF* measure and refers to a tweet in our context.

Thus, for the document d_j and the term t_i, the frequency of the term in the document is given by the following equation:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where $n_{i,j}$ stands for the number of occurrences of the term t_i in d_j. The denominator is the number of occurrences of all terms in the document d_j.

The IDF (*Inverse Document Frequency*) measures the importance of the term in the corpus. It is defined as follows:

$$IDF_i = log_2 \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

where $|D|$ stands for the total number of documents in the corpus and $|\{d_j : t_i \in d_j\}|$ is the number of documents having the term t_i.

Finally, the TD-IDF is obtained as follows:

$$TF\text{-}IDF_{i,j} = TF_{i,j} \times IDF_i$$

Nevertheless relying only on knowledge of the hierarchy in a cube does not always allow a good aggregation (i.e., corresponding to a real situation). For instance, the characteristics of the words in tweets are not necessarily the same in a State and in a City.

In [7], in a very different context, we proposed a new measure called TF-$IDF_{adaptative}$. This measure has been defined in order not to focus on the number of documents but rather to the number of documents for a specific class and take into account the level in the hierarchy. So in our case, this measure is well adapted for handling available hierarchies as it does not calculate the representative terms from the number of documents but rather from the desired class at a specific level. It is defined as follows:

$$TF_{i,j} - IDF_i^k = \frac{n_{i,j}}{\sum_k n_{k,j}} \times log_2 \frac{|E^k|}{|\{e_j^k : t_i \in e_j^k\}|}$$

where $|E^k|$ stands for the total number of elements of type k (in our example, $k = \{$*City*, *State*, *Country*$\}$) which corresponds to the level of the hierarchy that the decision maker wants to aggregate. $|\{e_j : t_i \in e_j\}|$ is relative to the number of elements of type k where the term t_i appears. Thus, we define $IDF_{adaptive}$ as follows:

$$IDF_i^{C_l} = log_2 \frac{m}{|\{C_l : t_i \in C_l\}|}$$

where m stands for the total number of communities. $|\{C_l : t_i \in C_l\}|$ is the number of communities C_l where the term t_i appears.

Actually, this adaptive measure can be easily generalized for taking into account the different level of the hierarchies as well as the information the user is interested in. Then it is possible to define a context $C = \langle n, l, type\rangle$ where n stands for one node in the hierarchy, l the level of the hierarchy the user would like to extract information and *type* corresponds to the type of information, i.e. element of a specific level (e.g. city) or tweets for this level. For a context C and a term t_i, the *Generalized IDF* is defined as follows:

$$Generalized\ IDF_{C,i,j \in type} = log_2 \frac{|type^l|}{f(type)}$$

where $|type^l|$ stands for the total number of elements of type *type* occurring in the scope of n at the level l of the hierarchy. $f(type)$ is defined as:

$$f(type) = \begin{cases} if\ type\ = Extension_D & |\{d_j : t_i \in Extension_D(n,l)\}| \\ else & |\{n_k : \exists t_i \in docs(n_{k_l})\ and\ n_{k_l} \in Ext(n,l)\}| \end{cases}$$

Depending on the value of type, the $f(type)$ function returns either the number of documents specified in the extension having the term t_i or the number of nodes having at least one document with the term t_i.

Finally, for a context C and a term t_i, the *Generalized TF-IDF* is defined as:

$$Generalized\ TF\text{-}IDF_{C,i,j \in type} = TF_{i,j} \times Generalized\ IDF_{i,j}$$

2.3 Illustration

In this section we illustrate how such a model can be useful for the decision maker. To extract the tweets related to the vocabulary used in MeSH, we focus on the tweets related to "Disease" and request Twitter by using all the terms of the corresponding hierarchy. We thus collected 1,801,310 tweets from Janurary 2010 to February 2011 having at least one term of the Disease hierarchy. Experiments were performed by using PostgreSQL 8.4 with the Pentaho Mondrian 3.20 environment. Visualization are done by using the visualization set of tools provided by Google are

Figure 3 shows the distribution of words *hepatitis*, *leukomia* and *pneunomia* over the period (excluding US). We can note that the word *pneunomia* appears a lot in the set of tweets in January 2009 and in February 2010. By using visualization tools it is possible to visualize the worldwide coverage of this disease (Figure 4). This coverage is obtained by fixing the location dimension and by examining the countries for which this word has been selected as more relevant for a country with our measures.

In Figure 4 we can notice that *pneunomia* has been extensively reported in tweets from Russia, India, Ecuador or Australia. News available at that period might be useful to better understanding this behavior. For instance, in the second week of January 2001, the British singer Trish Keenan died of pneumonia after contracting swine flu in Australia[6] and, in Ecuador, an alarm over cases of severe pneumonia similar to the H1N1 virus was triggered after that the Merced hospital has received 35 cases, 18 occurred in the first week of January, and reported that two people died. and just as serious[7]. In Russia also, In December of 2010, in

[6] http://www.dailymail.co.uk/news/article-1347160/

[7] http://www.flutrackers.com/forum/showthread.php?t=158136

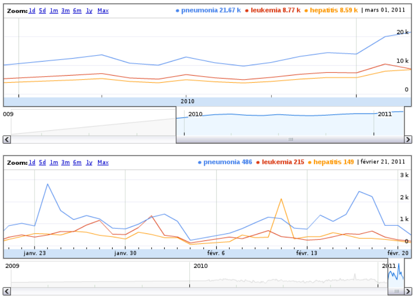


Fig. 3. Distribution of the use of words pneumonia, hepatitis and leukemia over the period (top) and for January and February (bottom)

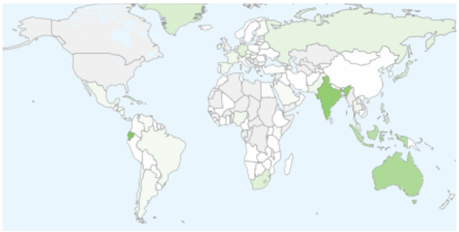
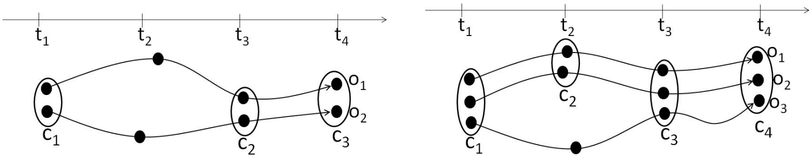


Fig. 4. Distribution of the use of the word pneunomia



(a) Swarm

(b) Convoy

Fig. 5. An example of swarm and convoy where c_1, c_2, c_3, c_4 are clusters

Yurga /Kemerovo region, over 200 soldiers were taken to hospitals with a bad cold and several people were in critical condition: severe pneumonia[8]. All these events were extensively tweeted or re-tweeted in these countries.

3 Pattern Mining for Tweets

Usually pattern mining approaches focus on extracting different kinds of patterns (i.e. itemsets, sequences, trees, graphs, etc) hidden in the database. Using these patterns for analyzing tweets is one of the topic addressed by some research work. In this paper we investigate another kind of patterns coming from a very different context: spatio-temporal patterns. Our main objective is to highlight that knowledge extracting can be very useful for the decision maker. Basically, spatio-temporal patterns are defined in a totally different context (i.e spatio-temporal data) and aim to identify groups of moving objects for which a strong relationship and interaction exist within a defined spatial region during a given time duration. Recently, many patterns have been defined such as flocks [11], convoys [12], swarms, closed swarms [13], moving clusters [14], group pattern [15], etc.

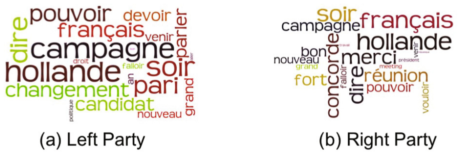
Let us assume that we have a group of moving objects $O_{DB} = \{o_1, o_2, \ldots, o_z\}$, a set of timestamps $T_{DB} = \{t_1, t_2, \ldots, t_n\}$ and at each timestamp $t_i \in T_{DB}$,

[8] http://www.flutrackers.com/forum/showthread.php?t=156585

spatial information[9] x, y for each object. . Usually, in spatio-temporal mining, we are interested in extracting a group of objects staying together during a period of time. Therefore, from now, $O = \{o_{i_1}, o_{i_2}, \ldots, o_{i_p}\} (O \subseteq O_{DB})$ stands for a group of objects, $T = \{t_{a_1}, t_{a_2}, \ldots, t_{a_m}\}$ $(T \subseteq T_{DB})$ is the set of timestamps within which objects stay together. Let ε be a user-defined threshold standing for a minimum number of objects and min_t a minimum number of timestamps. Thus $|O|$ (resp. $|T|$) must be greater than or equal to ε (resp. min_t). Here we focus on two particular patterns to illustrate. Informally, a *swarm* is a group of moving objects O containing at least ε individuals which are closed each other for at least min_t timestamps. For example, as shown in Figure 5a, if we set $\varepsilon = 2$ and $min_t = 2$, we can find the following swarms $(\{o_1, o_2\}, \{t_1, t_3\})$, $(\{o_1, o_2\}, \{t_1, t_4\})$, $(\{o_1, o_2\}, \{t_3, t_4\})$, $(\{o_1, o_2\}, \{t_1, t_3, t_4\})$. We can also note that these swarms are in fact redundant since they can be grouped together in the following closed swarm $(\{o_1, o_2\}, \{t_1, t_3, t_4\})$. A *convoy* is also a group of objects such that these objects are closed each other during at least min_t *consecutive* time points. For instance, on Figure 5, with $\varepsilon = 2, min_t = 2$ we have two convoys $(\{o_1, o_2\}, \{t_1, t_2, t_3, t_4\})$ and $(\{o_1, o_2, o_3\}, \{t_3, t_4\})$. Recently in [16] we proposed GET_MOVE an unifying approach for extracting all these kinds of patterns. Furthermore in [17], we proposed new kinds of patterns, called gradual patterns. For instance, they can be useful to extract gradual trajectories such as: "From October to December the more time passes, the more Eagle are moving from Canada to Mexico" or "From June to July, the more the time goes by, the more people are going to Miami". All these approaches share the same pre-processing: a clustering algorithm is applied (e.g. DBSCAN) for extracting clusters grouping together objects closed to the same location. During our research we considered that the clustering can be applied in other dimensions. Recently, we defined a new project called POLOP[10] (*Political Opinion Mining*) which aims to cope with the analysis of the evolution of French political communities over Twitter during 2012 both in terms of relevant terms, opinions, behaviors. 2012 is particularly important for French political communities dues the two main elections: Presidential and Legislative. From the 12th December 2011 to the 19th June 2012, we thus obtained 2,122,012 tweets from 213,005 users. For 130,618 tweets, 232 users can unambiguously be assigned to a political party (i.e. user is a politician or an official political community account). By using our defined measures (C.f. Section 2.2), we can select for different political parties the set of relevant words at different periods (see Figure 6). In order to extract interesting trajectories for different parties we applied a clustering technique, for each party, on the top-k set of relevant terms over time. Thus, we group together users from the same party sharing the same words. Figure 7 illustrates a kind of trajectory that might be extracted from the political tweets. In February 2012, in one of the first debate, the focus of the French election campaign suddenly shifts from the economy to racism and national identity. For the right party, as we can notice, during the debate, most of used words are such that lots of party members shared the same words. Just after the debate, some right

[9] Spatial information can be, for instance, GPS location.

[10] `http://www.lirmm.fr/~bouillot/polop/polop.html`



(a) Left Party

(b) Right Party

Fig. 6. An example of the most relevant words of parties: (a) Left Party and (b) Right Party

party users move to another cluster where words such as "national identity", "France" while other ones shift to other topic like "elections". Actually, during the debate the main trend was to motivate some electors from the extreme right. During this time, as we can notice, in the left party, as it is illustrated through the gradual pattern, the left party uses completely different kinds of words but just when right users behavior split.

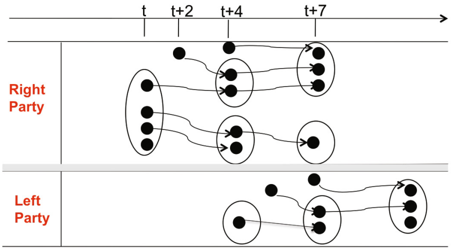


Fig. 7. An example of a trajectory that can be extracted from tweets

4 Conclusions

In this paper we first proposed a new approach to analyze tweets from their multidimensional characteristics. The originality of our proposal is to define and manipulate cubes of tweets. We have shown that the analysis of tweets requires the definition of new measures and that a contextualization step is relevant. Then we investigated how pattern mining and more precisely trajectories can be useful for analyzing tweet user behaviors. Future work involves several issues. First we want to extend the proposed approach to take into account opinions or feelings expressed in the tweets. Furthermore, as too many trajectories can be extracted from tweets, we would like to investigate how the minimal description length can be used to report only the most relevant patterns, i.e. top-k patterns.

References

1. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In: Proceedings of WWW, pp. 851–860 (2010)
2. Mathioudakis, M., Koudas, N.: Twittermonitor: trend detection over the twitter stream. In: Proceedings of SIGMOD 2010, pp. 1155–1158 (2010)

3. Li, C., Sun, A., Datta, A.: Twevent: Segment-based event detection from tweets. In: Proceedings of CIKM (2012)
4. Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., Lee, B.S.: Twiner: Named entity recognition in targeted twitter stream. In: Proceedings of SIGIR (2012)
5. Tsolmon, B., Kwon, A.-R., Lee, K.-S.: Extracting social events based on timeline and sentiment analysis in twitter corpus. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 265–270. Springer, Heidelberg (2012)
6. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Proceedings of COLING (2010)
7. Bringay, S., Béchet, N., Bouillot, F., Poncelet, P., Roche, M., Teisseire, M.: Towards an on-line analysis of tweets processing. In: Hameurlain, A., Liddle, S.W., Schewe, K.-D., Zhou, X. (eds.) DEXA 2011, Part II. LNCS, vol. 6861, pp. 154–161. Springer, Heidelberg (2011)
8. Codd, E., Codd, S., Salley, C.: Providing olap (on-line analytical processing) to user-analysts: An it mandate. White Paper, pp. 3–5 (1993)
9. Pérez-Martínez, J.M., Llavori, R.B., Cabo, M.J.A., Pedersen, T.B.: Contextualizing data warehouses with documents. Decision Support Systems 45(1), 77–94 (2008)
10. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
11. Vieira, M., Bakalov, P., Tsotras, V.: On-line discovery of flock patterns in spatio-temporal data. In: Proceedings of SIGSPATIAL (2009)
12. Jeung, H., Yiu, M., Zhou, X., Jensen, C.S., Shen, H.: Discovery of convoys in trajectory databases. PVLDB 1 (2008)
13. Li, Z., Ji, M., Lee, J.G., Tang, L., Yu, Y., Han, J., Kays, R.: Movemine: Mining moving object databases. In: Proceedings of SIGMOD (2010)
14. Jensen, C., Lin, D., Ooi, B.: Continuous clustering of moving objects. IEEE TKDE (2007)
15. Wang, Y., Lim, E.P., Hwang, S.Y.: Efficient mining of group patterns from user movement data. DKE (2006)
16. Nhat Hai, P., Poncelet, P., Teisseire, M.: GET_MOVE: An efficient and unifying spatio-temporal pattern mining algorithm for moving objects. In: Hollmén, J., Klawonn, F., Tucker, A. (eds.) IDA 2012. LNCS, vol. 7619, pp. 276–288. Springer, Heidelberg (2012)
17. Nhat, H.P., Ienco, D., Poncelet, P., Teisseire, M.: Mining time relaxed gradual moving object clusters. In: Proceedings of SIGSPATIAL (2012)
18. Landauer, T.K., Foltz, P.W., Laham, D.: Introduction to latent semantic analysis. Discourse Processes 25 (1998)
19. Turney, P.D.: Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 491–502. Springer, Heidelberg (2001)
20. Tang, J., Jin, R., Zhang, J.: A topic modeling approach and its integration into the random walk framework for academic search. In: Proceedings of ICDM (2008)

Weighted Line Graphs for Overlapping Community Discovery and their Evaluation

Tetsuya Yoshida

Graduate School of Information Science and Technology,
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
yoshida@meme.hokudai.ac.jp

Abstract. Community discovery has often been achieved by assigning each node in a network only to one community. However, a node (e.g., user) might belong to several communities in real world networks. For undirected connected networks without self-loops, we proposed weighted line graphs based on the weights of the original network, as they do not contain self-loops as in the standard line graph in general graph theory. Overlapping community discovery is achieved by applying some off-the-shelf node partitioning method to the weighted line graphs. In this paper we report a performance evaluation of the weighted line graphs over both synthetic and real-world networks. The effectiveness of the weighted line graphs are investigated in terms of both the visualization of discovered communities and the generalized modularity measure. The results show that both the utilization of weights in the original networks and the self-loop free property contribute to the performance improvement.

1 Introduction

Community discovery from networks is usually defined as a task of finding out groups of nodes in a network as communities [10,9,7,8]. Thus, node partitioning of a network is conducted in most previous approaches. However, in real world networks, a node might belong to more than one community. For instance, an individual, which is represented as a node in a network, can belong to several interested groups in social networks depending on his/her favorite genre of music, films, games, etc. Thus, node partitioning of a network can lead to miss some important aspects of real-world networks.

By transforming the original network into the corresponding line graph, overlapping community discovery is achieved by applying some off-the-shelf node partitioning method to the transformed graph. We proposed an approach for overlapping community discovery, and preliminary results are reported as a poster paper in [12]. For undirected connected networks without self-loops, weighted line graphs are constructed based on the weights of the original network, as they do not contain self-loops as in the standard line graph in general graph theory.

In this paper we report a performance evaluation of our weighted line graphs in [12] over both synthetic and real-world networks. Extensive experiments are

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 121–130, 2013.

conducted, and the effectiveness of the weighted line graphs are analyzed in terms of both the visualization of the discovered communities and a generalized modularity measure. The results indicate that our weighted line graphs can improve the quality of the discovered overlapping communities. The results are encouraging, and show that both the utilization of weights in the original networks and the self-loop free property contribute to the performance improvement.

Preliminaries: We use a bold italic lowercase letter to denote a vector, and a bold normal uppercase letter to denote a matrix, and $\mathbf{X}^T$ stands for the transposition of a matrix $\mathbf{X}$. $\mathbf{X}_{ij}$ stands for the element in a matrix $\mathbf{X}$. $\mathbf{1}_n \in \mathbb{R}^n$ stands for a column vector where each element is 1.

A network (graph) G=(V, E) consists of a set of nodes (vertices) V and a set of links (edges) E[1]. Since most social networks are represented simple graphs (undirected graphs without self-loops) [6], we focus on this type of networks. Let n stands for the number of nodes in G, and m for the number of links in G.

The connectivity of a network can be represented as a square matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, which is called an adjacency matrix. $\mathbf{A}_{ij} = 1$ if the pair of nodes (v_i, v_j) is connected; otherwise, $\mathbf{A}_{ij} = 0$. For a simple graph, the corresponding adjacency matrix $\mathbf{A}$ is symmetric and its diagonal elements are zeros. The vector $\boldsymbol{k} = \mathbf{A}\mathbf{1}_n$ denotes the degree vector, where k_i represents the degree (number of adjacent links) of node i. For a network G, an incidence matrix $\mathbf{B} \in \{0,1\}^{n \times m}$ is defined as: $\mathbf{B}_{i\alpha}$ =1 if a link α is related to a node i in G; otherwise, $\mathbf{B}_{i\alpha}$ =0 [3].

2 Weighted Line Graphs

2.1 Previous Weighted Line Graphs

Several kinds of weighted line graphs were proposed based on the adjacency matrix of the network in [4]. Originally, the line graph of a simple graph is defined in the general graph theory. For a simple graph G, the adjacency matrix $\mathbf{C}$ of the corresponding line graph $L(G)$ in general graph theory can be represented in terms of the incidence matrix $\mathbf{B}$ of G as:

$$\mathbf{C} = \mathbf{B}^T\mathbf{B} - 2\mathbf{I}_m \tag{1}$$

where $\mathbf{I}_m \in \{0,1\}^{m \times m}$ stands for the square identity matrix of size m.

Several weighted line graphs were proposed in the previous approach [4]. The following weighted adjacency matrices were proposed for weighted line graphs:

$$\mathbf{E} = \mathbf{B}^T\mathbf{D}^{-1}\mathbf{B} \tag{2}$$

$$\mathbf{E}_1 = \mathbf{B}^T\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}\mathbf{B} \tag{3}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = k_i$[2]. The diagonal matrix $\mathbf{D}$ is also represented as $diag(\boldsymbol{k})$ in this paper.

[1] We also call a network as a graph, a node as a vertex, and a link as an edge.

[2] The i-th diagonal element is set to the i-th element of $\boldsymbol{k}$ in $\mathbf{D}$.

2.2 Weighted Line Graphs for Weighted Networks

Suppose a simple graph G (with n nodes and m links) contains non-negative weights on links, and w_{ij} stands for the weight on the link between node i and node j. We assume that the weight w_{ij} can be interpreted as the similarity between the nodes [11,13].

Representation Matrices for Weighted Networks. We generalized the representation matrices of the original network from 0-1 matrices to weighted ones in [12]. Let $\tilde{\mathbf{A}}$ stands for a weighted adjacency matrix of a network G where $\tilde{\mathbf{A}}_{ij} = w_{ij}$. In addition, we define a weighted incidence matrix $\tilde{\mathbf{B}}$ based on the weights in G. For a link α=(i, j) which is adjacent to node i and node j in G, $\tilde{\mathbf{B}}_{i\alpha}$ and $\tilde{\mathbf{B}}_{j\alpha}$ are set to the weight w_{ij} in G; other elements in the α-th column of $\tilde{\mathbf{B}}$ are set to zeros. Thus, as in the generalization from $\mathbf{A}$ to $\tilde{\mathbf{A}}$, the standard 0-1 incidence matrix $\mathbf{B}$ is generalized to $\tilde{\mathbf{B}}$ based on the weights in G.

Based on $\tilde{\mathbf{A}}$, we also generalize the degree matrix $\boldsymbol{k}$ and the diagonal matrix $\mathbf{D}$ as: $\tilde{\boldsymbol{k}} = \tilde{\mathbf{A}}\mathbf{1}_n$, and $\tilde{\mathbf{D}} = diag(\tilde{\boldsymbol{k}})$. The vector $\tilde{\boldsymbol{k}}$ represents the sum of weights on links which are adjacent to each node in G. The matrix $\tilde{\mathbf{D}}$ corresponds to the weighted counterpart of the diagonal matrix $\mathbf{D}$ in eq.(2) and eq.(3).

Weighted Line Graphs for Weighted Networks. Based on the above matrices, we define the following representation matrices for weighted line graphs, which are the generalization of the standard line graph as well as the ones in [4]:

$$\tilde{\mathbf{C}} = \tilde{\mathbf{B}}^T\tilde{\mathbf{B}} - 2diag(\boldsymbol{w'}) \tag{4}$$

$$\tilde{\mathbf{E}} = \tilde{\mathbf{B}}^T\tilde{\mathbf{D}}^{-1}\tilde{\mathbf{B}} \tag{5}$$

$$\tilde{\mathbf{E}}_1 = \tilde{\mathbf{B}}^T\tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1}\tilde{\mathbf{B}} \tag{6}$$

where $\boldsymbol{w'} = \boldsymbol{w} \odot \boldsymbol{w}$. Here, $\odot$ stands for the Hadamard product (element-wise product) [5].

Theorem 1. *Following properties hold for the previous weighted line graphs and our weighted line graphs:*

$$\mathbf{1}_m^T\mathbf{C} = (\boldsymbol{k} - \mathbf{1}_n)^T\mathbf{B}, \qquad \mathbf{1}_m^T\tilde{\mathbf{C}} = (\tilde{\boldsymbol{k}} - \mathbf{1}_n)^T\tilde{\mathbf{B}} \tag{7}$$

$$\mathbf{1}_m^T\mathbf{E} = 2\mathbf{1}_m^T, \qquad \mathbf{1}_m^T\tilde{\mathbf{E}} = 2\boldsymbol{w}^T \tag{8}$$

$$\mathbf{1}_m^T\mathbf{E}_1 = 2\mathbf{1}_m^T, \qquad \mathbf{1}_m^T\tilde{\mathbf{E}}_1 = 2\boldsymbol{w}^T \tag{9}$$

Proofs are omitted due to space shortage.

2.3 Weighted Line Graphs without Self-loops

For a simple graph, the corresponding line graph in general graph theory is also a simple graph. However, previous representation matrices $\mathbf{E}$ and $\mathbf{E}_1$ contain self-loops (i.e., diagonal elements are not zeros). Thus, since the standard line graph in general graph theory does not contain self-loops, the structural correspondence of the original network and its transformed graph did not hold.

Suppose $\mathbf{M} \in \mathbb{R}_{+}^{\ell \times \ell}$ is a (weighted) adjacency matrix for a network with non-zero diagonal elements (i.e., with self-loops). Our method distributes the diagonal elements of $\mathbf{M}$ into off-diagonal elements while sustaining some properties of $\mathbf{M}$. To realize this, we define the following matrices and vectors:

$$\boldsymbol{m} = diag(\mathbf{M}) \tag{10}$$

$$\mathbf{D}_M = diag(\boldsymbol{m}) \tag{11}$$

$$\mathbf{M}_{wo} = \mathbf{M} - \mathbf{D}_M \tag{12}$$

$$\boldsymbol{m}_{wo} = \mathbf{M}_{wo}\mathbf{1}_\ell \tag{13}$$

Based on the above matrices and vectors, our method in [12] transforms the matrix $\mathbf{M} \in \mathbb{R}_{+}^{\ell \times \ell}$ into the following matrix $\mathbf{N} \in \mathbb{R}_{+}^{\ell \times \ell}$ as:

$$\mathbf{N} = \mathbf{M}_{wo} + \mathbf{D}_M^{1/2}\ diag(\boldsymbol{m}_{wo})^{-1/2}\ \mathbf{M}_{wo}\ diag(\boldsymbol{m}_{wo})^{-1/2}\ \mathbf{D}_M^{1/2} \tag{14}$$

where $diag(\boldsymbol{m}_{wo})$ is a diagonal matrix with $\boldsymbol{m}_{wo}$ (as $\mathbf{D}_M$ in eq.(11)).

The following properties hold for the transformed matrix $\mathbf{N}$:

Theorem 2. *For a symmetric square matrix* $\mathbf{M}$ *with non-negative real values, the following properties for the matrix* $\mathbf{N}$ *with eq.(14):*

$$diag(\mathbf{N}) = \mathbf{0}_\ell \tag{15}$$

$$\mathbf{N}^T = \mathbf{N} \tag{16}$$

$$\mathbf{N}\mathbf{1}_\ell = \mathbf{M}\mathbf{1}_\ell \tag{17}$$

Proofs are omitted due to space shortage.

By substituting the matrices $\mathbf{E}$ and $\mathbf{E}_1$ in [4] (and, our $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{E}}_1$ in eq.(5) and eq.(6)) as a symmetric matrix $\mathbf{M}$ in eq.(14), we can define the corresponding matrices $\mathbf{F}$ and $\mathbf{F}_1$ (and, $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{F}}_1$) without self-loops. Especially, from eq.(8), eq.(9) and eq.(17), the following properties hold.

Collorary 3. *The following properties hold for the adjacency matrices of weighted line graphs:*

$$\mathbf{1}_m^T\mathbf{F} = \mathbf{1}_m^T\mathbf{E} = 2\mathbf{1}_m^T, \qquad \mathbf{1}_m^T\tilde{\mathbf{F}} = \mathbf{1}_m^T\tilde{\mathbf{E}} = 2\boldsymbol{w}^T \tag{18}$$

$$\mathbf{1}_m^T\mathbf{F}_1 = \mathbf{1}_m^T\mathbf{E}_1 = 2\mathbf{1}_m^T, \qquad \mathbf{1}_m^T\tilde{\mathbf{F}}_1 = \mathbf{1}_m^T\tilde{\mathbf{E}}_1 = 2\boldsymbol{w}^T \tag{19}$$

2.4 An Illustrative Example

Let's consider community discovery from a weighted graph G_0 in Fig. 1. The number over the links in G_0 represents the weights, which corresponds to the similarities between nodes. In previous approach, the weighted graph G is treated as an unweighted graph G_1, and the adjacency matrix (e.g., $\mathbf{E}$ in the lower left of Fig. 1) of the corresponding $L(G)$ is defined based on G_1. On the other hand, our approach directly defines the weights over the line graph (e.g., $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{F}}$ in the lower right of Fig. 1) by reflecting the weights on G_0. Furthermore, contrary to $\mathbf{E}$ and $\tilde{\mathbf{E}}$, $\tilde{\mathbf{F}}$ does not contain any self-loops, and is isomorphic to the standard line graph (i.e., $\mathbf{C}$ in Fig. 1).

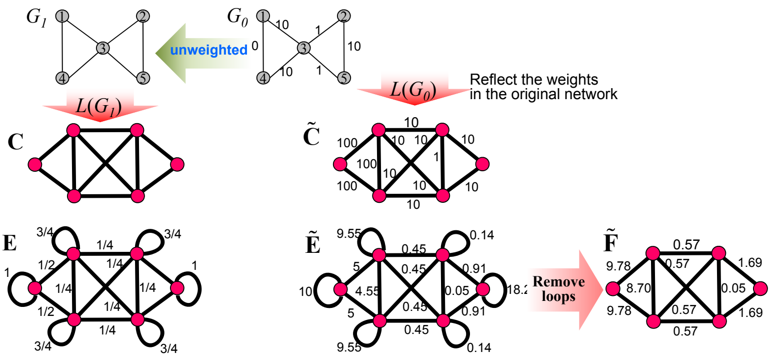


Fig. 1. Weighted line graphs of a weighted network

3 Evaluation

3.1 Modularity for Overlapping Community Discovery

For overlapping community discovery, we generalized the standard modularity for node partitioning [2]. The key idea is to generalize the conventional 0-1 indicator matrix to $\in \mathbb{R}_{+}^{n \times c}$ so that it represents a soft assignment of nodes into communities, as in fuzzy or probabilistic clustering. Thus, $\tilde{\mathbf{S}}$ needs to satisfy both $\tilde{\mathbf{S}}_{ij} \geq 0, \quad \forall i, j$ (each element is non-negative), and $\tilde{\mathbf{S}}\mathbf{1}_c = \mathbf{1}_n$(the assignments of a node to communities should sum up to 1)[3]. In an extremal case, each row of $\tilde{\mathbf{S}}$ can represent a uniform distribution of communities in the assignment of the node. The other extremal case is where $\tilde{\mathbf{S}}$ is a 0-1 matrix and represents the conventional hard assignment of nodes.

With $\tilde{\mathbf{S}}$, we proposed the following soft modularity Q_s [12]:

$$Q_s = \frac{1}{\mathbf{1}_n^T \mathbf{A} \mathbf{1}_n} \operatorname{tr}(\tilde{\mathbf{S}}^T (\mathbf{A} - \mathbf{P}) \tilde{\mathbf{S}}) \tag{20}$$

where $\mathbf{P}_{ij} = k_i k_j / 2m$. The larger Q_s is, the better the overlapping communities are specified in $\tilde{\mathbf{S}}$, as in the standard modularity measure [2].

Soft Indicator Matrix for Link Partitioning. Applying some off-the-shelf node partitioning method to our weighted line graphs results in a link partitioning of the original network. In link partitioning, each link is assigned to a community. Based on the community labels of links, we define a matrix $\tilde{\mathbf{H}}$ $\in R_{+}^{m \times c}$ as:

$$\tilde{\mathbf{H}}_{\alpha k} = \begin{cases} w_\alpha & \textit{if} \quad \textit{link}\ \alpha \text{ (with weight } w_\alpha\text{) is assigned to community } k \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

[3] c is the number of communities.

Based on $\tilde{\mathbf{H}}$ in eq.(21), we construct a soft indicator matrix $\tilde{\mathbf{S}}$, which represents a probabilistic assignment of nodes to communities, as:

$$\tilde{\mathbf{S}}_{ik} = \frac{\sum_{\alpha \text{ adjacent to } i} \tilde{\mathbf{H}}_{\alpha k}}{\sum_{k'} \sum_{\alpha \text{ adjacent to } i} \tilde{\mathbf{H}}_{\alpha k'}} \tag{22}$$

3.2 Weighted Networks

Synthetic Networks. We constructed each synthetic network by embedding communities into the overall network[4]. Both the overall network and communities were generated based on the Barabási-Albert iBAj model [1], which is known to have the scale-free degree distribution (as in the Internet). Let c stands for the number of communities, n_c for the number of nodes in a community (each network has $n_c \times c$ nodes). Let w_u stands for the link weight in the overall network, and $r_m > 1$ stands for the weight ratio for the links within communities.

Each synthetic network was generated as follows:

Step 1 : The overall network with $n_c \times c$ nodes was created with BA model. The constructed overall network was rather dense [5], and all the link weights were set to small w_u.

Step 2 : A network of n_c nodes was created for each community with BA model. In this case, the constructed communities were rather sparse [6], and all the link weights in the communities were set to $w_u \times r_m$ (i.e., $> w_u$).

Step 3 : The communities constructed at **Step 2** were embedded into the diagonal blocks of the adjacency matrix of the overall network at **Step 1**. Note that there was no overlap between the embedded diagonal blocks (i.e., embedded communities).

Step 4 : For each node i (with degree k_i) in the overall network, randomly select another community for which the node does not belong to. Then, randomly select up to k_i nodes in the selected community. Finally, connect the node i with the selected nodes with link weight $w_u \times r_m$.

The overall dense network with relatively small weights is constructed at **Step 1**. Sparse communities with large weights at **Step 2** are embedded into the overall network at **Step 3**. In addition, since each node is connected to other nodes in another community at **Step 4**, the constructed network has overlapping communities. In the following experiments, $w_u = 1$ and r_m=100 were used so that nodes in each community are tightly connected with large weights.

[4] `http://www-personal.umich.edu/~mejn/netdata/`
Pascal: `http://analytics.ijs.si/~blazf/pvc/data.html`
IV'04: `http://iv.slis.indiana.edu/ref/iv04contest/`

[5] The initial degree in BA model was set to 20 so that the underlying network is rather densely connected.

[6] The initial degree in BA model was set to 2 (1/10 of 20 in **Step 1**.)

Real-World Networks. We collected several weighted networks, and the largest connected component (LCC) of each network was utilized in the following experiments. Utilized networks are shown in Table 1 (the number of nodes and links in LCC are shown in the tables). The co-authorship relations is represented as an adjacency matrix $\tilde{\mathbf{A}}$, where $\tilde{\mathbf{A}}_{ij}$ stands for the number of papers which are coauthored by authors (nodes) i and j.

Table 1. Weighted Networks

dataset	#nodes	#links
lesmis	77	254
celegans	297	2148
Pascal	65	114
IV'04	112	255

3.3 Experimental Settings

Community Discovery Methods. The following off-the-shelf community discovery methods were utilized in the experiments: 1) labelPropagation [10], 2) leadingEigenvector [7], 3) walktrap [9].

LabelPropagation[10] works by labeling the vertices with unique labels and then updating the labels by majority voting in the neighborhood of the vertex.

Walktrap [9] tries to find densely connected subgraphs via random walks in the network. A metric based on short random walks is utilized to conduct the agglomerative clustering (partition) of a network into communities.

LeadingEigenvector [7] utilizes the eigenvector with the largest positive eigenvalue of **A** - **P**. Repeated bisection of a network is conducted in this method.

Since all of them conduct node partitioning of a network, these were utilized to conduct link partitioning with respect to the weighted line graphs.

Quality Measures. In each method, we evaluated the proposed Q_s in eq.(20). The soft indicator matrix in eq.(22) was constructed based on the node partitioning of line graphs by the above methods. Communities with larger value of Q_s are considered as better in terms of the modularity.

Note that the community discovery methods in Section 3.3 can return different communities even for the same network. Thus, we report the average of 10 runs for the same network.

Types of Line Graphs. We evaluated weighted line graphs based on the following representation matrices of each original network:

- 0-1 matrix [4]: $\mathbf{E}$ in eq.(2), $\mathbf{E}_1$ in eq.(3)
- weighted matrix: $\tilde{\mathbf{E}}$ in eq.(5), $\tilde{\mathbf{E}}_1$ in eq.(6)
- 0-1 matrix, without self-loops in eq.(14): $\mathbf{F}$ in eq.(18), $\mathbf{F}_1$ in eq.(19)
- weighted matrix, without self-loops in eq.(14): $\tilde{\mathbf{F}}$ in eq.(18), $\tilde{\mathbf{F}}_1$ in eq.(19)

3.4 Results of Synthetic Networks

By setting n_c (the number of nodes per community) to 50, synthetic networks were constructed by varying the number of communities. Since BA model is a

Table 2. Results of synthetic weighted networks

c	method	$\mathbf{E}$	$\tilde{\mathbf{E}}$	$\mathbf{F}$	$\tilde{\mathbf{F}}$	$\mathbf{E}_1$	$\tilde{\mathbf{E}}_1$	$\mathbf{F}_1$	$\tilde{\mathbf{F}}_1$
3	labelPropagation	0.001	0.063	0.025	**0.294**	0.000	0.102	0.000	0.045
	leadingEigenvector	0.025	**0.171**	0.025	0.170	0.029	0.134	0.030	0.135
	walktrap	0.057	0.279	0.064	0.353	0.073	0.365	0.073	**0.378**
4	labelPropagation	0.001	0.061	0.024	**0.289**	0.000	0.087	0.000	0.053
	leadingEigenvector	0.024	0.155	0.023	**0.156**	0.020	0.116	0.020	0.117
	walktrap	0.048	0.287	0.058	0.351	0.072	0.366	0.071	**0.387**
5	labelPropagation	0.001	0.059	0.024	**0.288**	0.000	0.063	0.000	0.027
	leadingEigenvector	0.023	**0.150**	0.023	0.148	0.015	0.100	0.015	0.100
	walktrap	0.048	0.276	0.053	0.353	0.074	0.355	0.077	**0.385**

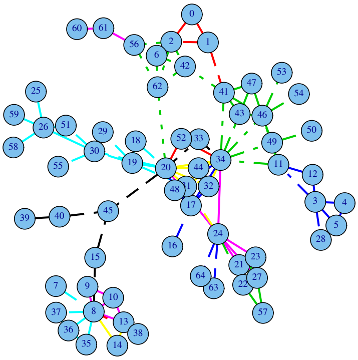


Fig. 2. Communities over $\mathbf{E}$ of Pascal with labelPropagation

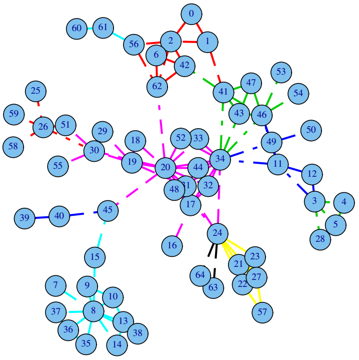


Fig. 3. Communities over $\tilde{\mathbf{F}}$ of Pascal with labelPropagation

random graph model, we generated 10 networks for the same value of c (the number of communities), and applied the community discovery methods 10 times. Thus, for synthetic networks, we report the average of 100 runs.

The results are shown in Table 2. In Table 2, each column corresponds to the adjacency matrix in Section 3.3, and each row corresponds to the community discovery methods in Section 3.3. For each row, the result with the maximal value of Q_s is shown in bold (the larger, the better). As shown in Table 2, our approach based on the weights of the original network and the self-loop free property (i.e., $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{F}}_1$) showed the best performance in almost all combinations of networks and the discovery methods.

Since the generated synthetic networks are sparse (in terms of the degree) within communities, if link weights are ignored, the communities are lost under the overall dense network. Thus, the previous approach (i.e.,$\mathbf{ECE}_1$ [4]) which does not reflect the weights could not capture the community structure, and the values of Q_s were very small. On the other hand, since our approach can reflect the weights in Section 2.2, the values of Q_s in our approach were almost 10 times larger than those of the previous approach. Furthermore, the self-loop free property in Section 2.3 was also effective.

Table 3. Results of real-world weighted networks

dataset	method	$\mathbf{E}$	$\tilde{\mathbf{E}}$	$\mathbf{F}$	$\tilde{\mathbf{F}}$	$\mathbf{E}_1$	$\tilde{\mathbf{E}}_1$	$\mathbf{F}_1$	$\tilde{\mathbf{F}}_1$
lesmis	labelPropagation	0.018	0.131	0.432	**0.449**	0.410	0.400	0.418	0.374
	leadingEigenvector	0.283	0.343	0.302	0.361	0.390	**0.426**	0.387	0.419
	walktrap	0.398	0.440	0.458	**0.495**	0.383	0.488	0.395	0.470
celegans	labelPropagation	0.004	0.084	0.057	**0.250**	0.000	0.059	0.000	0.032
	leadingEigenvector	0.111	0.174	0.112	0.179	0.207	**0.256**	0.207	**0.256**
	walktrap	0.343	0.343	0.335	0.362	0.305	0.316	0.305	**0.397**
Pascal	labelPropagation	0.059	0.091	0.460	**0.615**	0.572	0.591	0.609	0.607
	leadingEigenvector	0.432	0.467	0.437	0.474	0.459	0.503	0.549	**0.567**
	walktrap	0.528	0.535	0.627	**0.634**	0.571	0.574	0.569	0.575
IV04	labelPropagation	0.042	0.201	0.577	0.651	0.659	0.677	0.679	**0.700**
	leadingEigenvector	0.582	0.593	0.582	0.593	0.639	**0.646**	0.639	**0.646**
	walktrap	0.698	0.681	0.708	0.688	0.706	0.672	0.708	**0.722**

3.5 Results of Real-World Networks

As an example of overlapping community discovery, extracted communities for Pascal in Table 1 are shown in Fig. 2 and Fig.3. Fig. 2 corresponds to $\mathbf{E}$, and Fig.3 corresponds to our $\tilde{\mathbf{F}}$ (labelPropagation was applied to the corresponding weighted line graphs). In these figures, each community is depicted with the combination of link type and color. With our approach (Fig.3), the lower left subgraph (including node 8) and the central subgraph (including node 20) were not too much fragmented into multiple communities compared with the previous approach (Fig. 2).

The results are summarized in Table 3. Each column shows the values of Q_s in eq.(20). The largest value in each row is shown in bold. The results in Table 3 indicate that, by utilizing our weighted line graphs, it was possible to discover better communities by applying off-the-shelf node partitioning based methods. Compared with the previous matrices (i.e., $\mathbf{E}$ and $\mathbf{E}_1$), both the utilization of weights in the original networks (i.e., $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{E}}_1$) and the self-loop free property (i.e., $\mathbf{F}$ and $\mathbf{F}_1$, and $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{F}}_1$) outperformed in almost all networks and discovery methods.

Comparison of ($\mathbf{E}$, $\tilde{\mathbf{E}}$) and comparison of ($\mathbf{E}_1$, $\tilde{\mathbf{E}}_1$) indicate that the utilization of weights in the original networks contributes to the performance improvement in most cases. Similarly, comparison of ($\mathbf{E}$, $\mathbf{F}$) and comparison of ($\mathbf{E}_1$, $\mathbf{F}_1$) indicate that the self-loop free property contributes as well. When these are combined (i.e., $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{F}}_1$), the best performance was often achieved in Table 3.

4 Concluding Remarks

This paper reported a performance evaluation of weighted line graphs for overlapping community discovery. For undirected connected networks without self-loops, we proposed weighted line graphs based on the weights of the original network, as they do not contain self-loops as in the standard line graph in general graph theory. Overlapping community discovery is achieved by applying some off-the-shelf node partitioning method to the weighted line graph.

Through the experiments over both synthetic and real-world networks, the effectiveness of the weighted line graphs were investigated in terms of both the visualization of the discovered communities and the generalized modularity measure. The results are encouraging, and show that both the utilization of weights in the original networks and the self-loop free property contribute to the performance improvement. We plan to conduct more in-depth analysis of the discovered communities and extend our approach based on the analysis in near future.

Acknowledgments. This work is partially supported by Casio Science Promotion Foundation, Toyota Physical and Chemical Research Institute, and the Murata Science Foundation.

References

1. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
2. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E 70(6), 066111 (2004)
3. Diestel, R.: Graph Theory. Springer (2006)
4. Evans, T., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. Physical Review E 80(1), 016105:1–016105:8 (2009)
5. Harville, D.A.: Matrix Algebra From a Statistican's Perspective. Springer (2008)
6. Mika, P.: Social Networks and the Semantic Web. Springer (2007)
7. Newman, M.: Finding community structure using the eigenvectors of matrices. Physical Review E 76(3), 036104 (2006)
8. Newman, M.: Networks: An Introduction. Oxford University Press (2010)
9. Pons, P., Latapy, M.: Computing communities in large networks using random walks. Journal of Graph Algorithms 10(2), 191–218 (2006)
10. Raghavan, U., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Physical Review E 76, 036106 (2007)
11. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
12. Yoshida, T.: Overlapping community discovery via weighted line graphs of networks. In: Anthony, P., Ishizuka, M., Lukose, D. (eds.) PRICAI 2012. LNCS (LNAI), vol. 7458, pp. 895–898. Springer, Heidelberg (2012)
13. Yoshida, T.: Toward finding hidden communities based on user profile. Journal of Intelligent Information Systems (in press, 2013)

Graded Modal Logic GS5 and Itemset Support Satisfiability

Yakoub Salhi, Saïd Jabbour, and Lakhdar Sais

Université Lille-Nord de France
CRIL - CNRS UMR 8188
Artois, F-62307 Lens
{salhi,jabbour,sais}@cril.fr

Abstract. Graded modal logic GS5 is an extension of S5 by the modal connective $\Diamond_\lambda$: the formula $\Diamond_\lambda A$ means that there are at least λ worlds satisfying A. In this paper, we show how to reduce GS5 satisfiability to propositional satisfiability (SAT). Furthermore, we consider a satisfiability problem related to the frequent itemset mining problem: $\mathsf{SUPPSAT}^n$ (where n is a strictly positive integer). We show how $\mathsf{SUPPSAT}^n$ can be encoded in GS5 satisfiability and consequently in SAT.

1 Introduction

The modal logic S5 is among the most studied normal modal logics. In a possible world semantics, a model frame is a non empty set of *worlds* with an accessibility relation which is an equivalence relation [4]. However, there is another equivalent possible world semantics where a model frame is just a non empty set of worlds without any accessibility relation [9]. The formula $\Diamond A$ means simply that there exists a world where A is true and $\Box A$ means that A is true in every world. Let us note that there exist various formal systems dedicated to proof-search in this logic [9,22,15].

Graded modal logic GS5 includes modal connectives allowing to reason on the number of worlds satisfying a formula [7,21]. For example, the formula $\Diamond_5 A$ means that there are at least 5 worlds satisfying A. The first sound and complete axiomatization for this logic was provided by Fine in [7] and other results on axiomatic aspects were provided in [21]. Let us note that the modal connectives in GS5 coincide with concept cardinality restrictions in description logics defined in [3] and count operators in modal logics defined in [2].

In the literature, the majority of decision algorithms for modal logics are based on either the use of the formalism style in structural proof theory called sequent calculus and its variant called tableau method [8], or encodings in first order logic [16]. In the last decade, efficient SAT-based algorithms have been proposed to decide satisfiability of a formula in modal logics [10,11]. It has been formally proved that these algorithms are better than those based on tableau method [17]. One of the main ideas in the SAT-based approach consists simply in considering the modal subformulas as propositional variables and trying to

Y. Tanaka et al. (Eds.): ISIP 2012, CCIS 146, pp. 131–140, 2013.

extend propositional models to modal models. In this work, we are interested in finding reductions of satisfiability problem in S5 and GS5 to SAT.

After having introduced syntax and semantics of the modal logic S5, we present a reduction of S5 satisfiability problem to SAT. This reduction is obtained by using the property that says that a formula is satisfiable in S5 if and only if it is satisfiable in a model with at most the number of its modal subformulas as the number of worlds. Using a similar approach, we also show how to reduce GS5 satisfiability to SAT. Thus, efficient propositional SAT solvers can be used to perform satisfiability in S5 and its graded version GS5.

In order to show how GS5 satisfiability, and consequently SAT, can be used to reason over some problems in data mining, we consider a satisfiability problem in data mining: $\mathsf{SUPPSAT}^n$ (where n is a strictly positive integer) [5,6]. It is related to the frequent itemset mining problem which is one of the most studied problems in data mining [1]. It consists in computing frequent itemsets in a transaction database. We show how $\mathsf{SUPPSAT}^n$ can be encoded in GS5 satisfiability.The $\mathsf{SUPPSAT}^n$ problem has important applications. For instance, in the privacy preserving data mining where the $\mathsf{SUPPSAT}^n$ problem can be used for trying to reconstruct parts of the original database from data mining outcomes: *inverse data mining* [14] (see [5,6] for some details). Moreover, the $\mathsf{SUPPSAT}^n$ problem can be used to improve the pruning of infrequent candidates in frequent itemset mining algorithms.

2 Modal Logic S5

2.1 Syntax and Semantics

The set of propositional formulae of S5, denoted $\mathsf{Form}_{\mathsf{S5}}$, is inductively defined from a set of propositional variables, denoted Prop (we use $p, q, r, \ldots$ to range over Prop), by using the propositional connectives $\wedge$ and $\neg$ and the modal connective $\Diamond$. The other propositional connectives and the modal connective $\Box$ can be expressed using $\wedge$, $\neg$ and $\Diamond$ as follows: $A \vee B =_{def} \neg(\neg A \wedge \neg B)$, $A \rightarrow B =_{def} \neg A \vee B$ and $\Box A =_{def} \neg\Diamond\neg A$. In other words, the language of S5 extends that of classical propositional logic by the modal connectives $\Box$ and $\Diamond$.

A Hilbert axiomatic system for S5 is given by the following axioms and rules:
0. Any substitution instance of a propositional tautology.
$\mathcal{K}$. $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$ $\mathcal{T}$. $\Box A \rightarrow A$ $\mathcal{B}$. $A \rightarrow \Box\Diamond A$ 4. $\Box A \rightarrow \Box\Box A$

$$\frac{A \rightarrow B \qquad A}{B}\ [mp] \quad \text{and} \quad \frac{A}{\Box A}\ [nec]$$

Le us now define the simplest possible world semantics for S5. We first define the structure of S5 modal model, and then we define a relation between the worlds and $\mathsf{Form}_{\mathsf{S5}}$, called forcing relation, that allows us to define S5 satisfiability problem.

Definition 1 (Modal Model). *A modal model is a couple $\mathcal{M} = (W, V)$ where W is a non-empty set (of worlds) and V is a function from W to $\mathcal{P}$(Prop), where $\mathcal{P}$(Prop) is the powerset of Prop, i.e. its set of subsets.*

Definition 2 (Forcing Relation). *Let $\mathcal{M} = (W, V)$ be a modal model. The forcing relation, denoted $\vDash_{\mathcal{M}}$, between W and $\mathsf{Form}_{\mathsf{S5}}$ is inductively defined on formula structure as follows:*

$w \vDash_{\mathcal{M}} p$ *iff* $p \in V(w)$;

$w \vDash_{\mathcal{M}} A \wedge B$ *iff* $w \vDash_{\mathcal{M}} A$ *and* $w \vDash_{\mathcal{M}} B$;

$w \vDash_{\mathcal{M}} \neg A$ *iff* $w \nvDash_{\mathcal{M}} A$;

$w \vDash_{\mathcal{M}} \Diamond A$ *iff* $\exists w' \in W$, $w' \vDash_{\mathcal{M}} A$.

Definition 3 (S5 Satisfiability Problem). *A formula A is satisfiable in* S5 *iff there exists a modal model $\mathcal{M} = (W, V)$ sand a world $w \in W$ such that $w \vDash_{\mathcal{M}} A$.*

A formula A is valid in $\mathcal{M}$, denoted $\mathcal{M} \vDash A$, if and only if, for all $w \in W$, $w \vDash_{\mathcal{M}} A$. A is a theorem of S5 if and only if A is valid in every modal model. Satisfiability and validity are complementary in S5. In fact, S5 satisfiability (resp. validity) is NP-complete (resp. co-NP-Complete).

We define the size of a formula, denoted $|\cdot|$, as follows:

$$|p| = 1 \quad |\neg A| = |A| + 1 \quad |A \wedge B| = |A| + |B| + 1 \quad |\Diamond A| = |A| + 1$$

Theorem 1 ([13]). *A is satisfiable in* S5 *iff it is satisfiable in a model with at most $|A|$ worlds.*

2.2 Reducing S5 Satisfiability to SAT

We first introduce the satisfiability problem and some necessary notations that will be used in the different reductions of our problems to SAT. A *literal* is a positive (p) or negated ($\neg p$) propositional variable. The two literals p and $\neg p$ are called *complementary.* We denote by $\bar{l}$ the complementary literal of l. Let us recall that any propositional formula can be translated to the conjunctive normal form (CNF) using linear Tseitin encoding [20]. We denote by $Var(F)$ the set of propositional variables appearing in F. An *interpretation* $\mathcal{B}$ of a propositional formula F is a function which associates a value $\mathcal{B}(p) \in \{0, 1\}$ (0 corresponds to $false$ and 1 to *true*) to the variables $p \in Var(F)$. A *model* of a formula F is an interpretation $\mathcal{B}$ that satisfies the formula. The SAT problem consists in deciding if a given formula admits a model or not.

Let us now show how to reduce S5 satisfiability to SAT. We first associate to each S5 formula A and strictly positive integer n a propositional formula so that A is satisfiable in a model with n worlds if and only if its associated propositional formula is satisfiable. Thus, using the property that a formula is satisfiable in S5 if and only if it is satisfiable in a model with at most the number of its modal subformulas as the number of worlds, we provide a polynomial reduction of S5 satisfiability to SAT.

We define a *deep modal subformula* as a subformula of the form $\Diamond A$ that is in the scope of only one modal connective. For instance, the deep modal subformulas of $\Diamond(\Diamond A \wedge \Diamond(C \vee \Diamond D))$ are $\Diamond A$ and $\Diamond(C \vee \Diamond D)$.

Definition 4. *Let A be an* S5 *formula, $\{\Diamond B_1, \ldots, \Diamond B_k\}$ the set of the deep modal subformulas of A and $\{x_{\Diamond B_1}, \ldots, x_{\Diamond B_k}\}$ a set of k fresh variables. We define inductively the* S5 *formula $\mathcal{D}(A)$ as follows:*

- *$\mathcal{D}(A) = A$ if A does not contain any deep modal subformula;*
- *$\mathcal{D}(A) = A[\Diamond B_1/x_{\Diamond B_1}, \ldots, \Diamond B_k/x_{\Diamond B_k}] \bigwedge_{i=1}^{k} \mathcal{D}(\Diamond x_{\Diamond B_i} \rightarrow \Diamond B_i)$ $\bigwedge_{i=1}^{k} \mathcal{D}(\Diamond B_i \rightarrow \neg\Diamond\neg x_{\Diamond B_i})$.*

where $A[\Diamond B_1/x_{\Diamond B_1}, \ldots, \Diamond B_k/x_{\Diamond B_k}]$ denotes the result of substituting each deep modal subformula by its associated fresh variable.

It is easy to see that $\mathcal{D}(A)$ is of polynomial size in $|A|$. Intuitively, the two formulae $\Diamond x_{\Diamond B_i} \rightarrow \Diamond B_i$ and $\Diamond B_i \rightarrow \neg\Diamond\neg x_{\Diamond B_i}$ correspond to $\Box(x_{\Diamond B_i} \leftrightarrow \Diamond B_i)$. Indeed, we have $\Box(x_{\Diamond B_i} \rightarrow \Diamond B_i) \equiv \Box(\neg x_{\Diamond B_i} \vee \Diamond B_i) \equiv \neg\Diamond x_{\Diamond B_i} \vee \Box\Diamond B_i \equiv \Diamond x_{\Diamond B_i} \rightarrow \Diamond B_i$ (because of $\Box\Diamond B_i \equiv \Diamond B_i$ and $\Box\neg x_{\Diamond B_i} \equiv \neg\Diamond x_{\Diamond B_i}$). Furthermore, $\Box(\Diamond B_i \rightarrow x_{\Diamond B_i}) \equiv \Box\neg\Diamond B_i \vee \Box x_{\Diamond B_i} \equiv \neg\Diamond B_i \vee \neg\Diamond\neg x_{\Diamond B_i}$. Thus, one can easily prove that A is satisfiable if and only if $\mathcal{D}(A)$ is satisfiable. Moreover, we have for all model $\mathcal{M}$, if $\mathcal{M} \models \mathcal{D}(A)$ then $\mathcal{M} \models A$. In fact, it is well-known that one can disallow nested modalities in S5 without any loss of generality. Here, we use an approach similar to that using in Tseitin encoding in propositional logic [20].

Definition 5. *Let A be an* S5 *formula and n a strictly positive integer. We associate to each propositional variable p of A and $i \in \{1, \ldots, n\}$ a new propositional variable p_i. Then, we define the propositional formula $(A)^i_n$ by induction on the structure of A as follows:*

$$(p)^i_n = p_i \qquad (B \wedge C)^i_n = (B)^i_n \wedge (C)^i_n$$
$$(\neg B)^i_n = \neg(B)^i_n \qquad (\Diamond B)^i_n = \bigvee_{1 \leqslant j \leqslant n} (B)^j_n$$

Let us note that $(A)^i_n$ is not always of polynomial size in $|A|$ (we consider that n is polynomial in $|A|$). For instance, we have with a naive enconding:

$$(\overbrace{\Diamond \cdots \Diamond}^{k\ times} p)^i_n = \overbrace{(p_1 \vee \cdots \vee p_n) \vee \cdots \vee (p_1 \vee \cdots \vee p_n)}^{k^{n-1}\ times}$$

However, one can see that the propositional formula $(\mathcal{D}(A))^i_n$ is in the polynomial size of $|A|$. Moreover, $(\mathcal{D}(A))^i_n$ is satisfiable if and only if $(A)^i_n$ is satisfiable, and if an interpretation $\mathcal{B}$ is a model of $(\mathcal{D}(A))^i_n$ then it is also a model of $(A)^i_n$.

Proposition 1. *A is* S5 *satisfiable in a modal model of n worlds iff $(\mathcal{D}(A))^1_n$ is satisfiable.*

Proof. We only have to prove that A is S5 satisfiable in a modal model of n worlds iff $(A)^1_n$ is satisfiable.

Part $\Rightarrow$. Let $\mathcal{M} = (W, V)$ be a modal model with n worlds such that there exists w in W where $w \models_{\mathcal{M}} A$. Let f be a function associating to each integer $i \in \{1, \ldots, n\}$ a world in W such that $f(1) = w$ and for all $i, j \in \{1, \ldots, n\}$, if $i \neq j$ then $f(i) \neq f(j)$. We define the boolean interpretation $\mathcal{B}$ as follows: $\mathcal{B}(p_i)$ is equal to 1 if $p \in V(f(i))$, 0 otherwise. One can easily prove by simultaneous induction on formula structure that for all $F \in \mathsf{Form}_{\mathsf{S5}}$ and for all $i \in \{1, \ldots, n\}$:

- if $f(i) \models_{\mathcal{M}} F$ then $\mathcal{B}((F)^i_n) = 1$; and
- if $f(i) \not\models_{\mathcal{M}} F$ then $\mathcal{B}((F)^i_n) = 0$.

Therefore, we deduce that $\mathcal{B}$ is a model of $(A)^1_n$.

Part $\Leftarrow$. Let $\mathcal{B}$ be a model of $(A)^1_n$ and $\mathcal{M} = (\{1,\ldots,n\}, V)$ a modal model such that for all $i \in \{1,\ldots,n\}$, $V(i) = \{p \mid B(p_i) = 1\}$. Similarly to Part $\Rightarrow$, one can easily prove by simultaneous induction on formula structure that for all $F \in \mathsf{Form}_{\mathsf{S5}}$ and for all $i \in \{1,\ldots,n\}$:
- if $\mathcal{B}((F)^i_n) = 1$ then $i \vDash_{\mathcal{M}} F$; and
- if $\mathcal{B}((F)^i_n) = 0$ then $i \nvDash_{\mathcal{M}} F$.

Since $\mathcal{B}$ satisfies $(A)^1_n$, we deduce that $1 \vDash_{\mathcal{M}} A$, and consequently $\mathcal{M}$ satisfies A.

Theorem 2. *A is* S5 *satisfiable iff* $\bigvee_{1\leqslant i\leqslant |A|}(\mathcal{D}(A))^1_i$ *is satisfiable.*

Proof. It is a direct consequence of Theorem 1 and Proposition 1.

3 Graded Modal Logic GS5

3.1 Syntax and Semantics

The graded modal logic GS5 corresponds to an extension of S5. Indeed, in this logic the modal connective $\Diamond$ is generalized to a new one $\Diamond_\lambda$ where λ is a strictly positive integer. The set of formulae of GS5 ($\mathsf{Form}_{\mathsf{GS5}}$) is given by the following grammar: $A, B ::= p \mid \neg A \mid A \wedge B \mid \Diamond_\lambda A$.
Semantically, the formula $\Diamond_\lambda A$ is satisfied in a model if and only if there exist at least λ different worlds satisfying A. Thus, such a modal logic allows to reason on propertiy supports. Concerning the dual modal connective, $\Box_\lambda A$ is satisfied if and only if there exist less than λ distinct worlds satisfying $\neg A$. Like in the case of S5, GS5 satisfiability is NP-complete.

The definition of forcing relation in the case of $\Diamond_\lambda$ is as follows:

$$w \vDash_{\mathcal{M}} \Diamond_\lambda A \quad \text{iff there exist } \lambda \text{ distinct worlds } w_1,\ldots,w_\lambda \text{ in } W \text{ such that } w_1 \vDash_{\mathcal{M}} A,\ldots,w_\lambda \vDash_{\mathcal{M}} A.$$

The formulae size $|\cdot|$ is extended to $\mathsf{Form}_{\mathsf{GS5}}$ by $|\Diamond_\lambda A| = |A| + \lambda$.

Theorem 3 ([21]). *A is* GS5 *satisfiable iff it is satisfiable in a model with at most $|A|$ worlds.*

Definition 6. *Let A be a* GS5 *formula and n a strictly positive integer. The* GS5 *formula $s(A,n)$ is inductively defined on the structure of A as follows:*

$s(p,n) = p$ $\qquad s(B \wedge C, n) = s(B,n) \wedge s(C,n)$
$s(\Diamond_\lambda B, n) = \Diamond_\lambda s(B,n)$ $\qquad s(\neg p, n) = \neg p$
$s(\neg\neg B, n) = s(B,n)$ $\qquad s(\neg(B \wedge C), n) = s(\neg B, n) \vee s(\neg C, n)$
$s(\neg\Diamond_\lambda B, n) = \Diamond_{n-\lambda+1} s(\neg B, n)$

Proposition 2. *A is* GS5 *satisfiable in a model $\mathcal{M}$ with n worlds iff $s(A,n)$ is* GS5 *satisfiable in $\mathcal{M}$*

Proof. By induction on the structure of A.

3.2 Reducing GS5 Satisfiability to SAT

In the case of $\Diamond_\lambda$ in GS5 satisfiablity, we have to determine if a formula is satisfied in at least λ worlds. In our reduction (Definition 7), such counting argument is encoded using the well known Horn cardinality constraint $\sum_{j=1}^{n} x_j \geqslant \lambda$. It is important to note that this kind of constraints and its generalized form $\sum_{j=1}^{n} a_j x_j \geqslant \lambda$ (where a_j are positive integers) can be polynomialy encoded as a propositional formula in CNF [12]. As mentioned by J. P. Warners in [12], the first polynomial CNF expansion of Horn cardinality constraint is first proposed by Hooker in an unpublished note. The Hooker encoding of $\sum_{j=1}^{n} x_j \geqslant \lambda$ to CNF is obtained as follows:

$$\neg z_{ik} \vee x_i, \quad i = 1, \ldots, n \quad k = 1, \ldots, \lambda \tag{1}$$

$$\bigvee_{i=1}^{n} z_{ik}, \quad k = 1, \ldots, \lambda \tag{2}$$

$$\neg z_{ik} \vee \neg z_{ik'}, \quad i = 1, \ldots, n \quad k, k' = 1, \ldots, \lambda, k \neq k' \tag{3}$$

The two equations 2 and 3 encode the pigeon hole problem $PH_{\lambda,n}$, where λ is the number of pigeons and n is the number of holes (z_{ik} expresses that pigeon k is in hole i). The mapping between the models of $PH_{\lambda,n}$ and those of $\sum_{j=1}^{n} x_j \geqslant \lambda$ are obtained thanks to the equation 1. In this elegant transformation, the number of additional variables is $\lambda \times n$ and the number of clauses required is $\frac{1}{2}\lambda(n^2+n+2)$. Let us mention that in [12] (see Section 5), the equation 3 is written as follows:

$$\neg z_{ik} \vee \neg z_{jk}, \quad i, j = 1, \ldots, n, i \neq j \quad k = 1, \ldots, \lambda. \tag{4}$$

One can obviously check that the formulation given in [12], where the equation 3 is substituted by the equation 4 is not correct. Even the description associated to the formulation in Warners' paper is not correct. This brief recall allows us to give a correct formulation described by the three equations 1, 2 and 3. Several improvements of the CNF encoding of both Horn cardinality constraints (e.g. [19,18]) have been proposed since 1996. In these recent encodings the propagation capabilities of the obtained CNF has been significantly enhanced.

Let us now introduce our reduction of GS5 satisfiability to SAT. To this end, we use an approach similar to that used in the case of S5 satisfiability.

Definition 7. *Let A be a* GS5 *formula and n a strictly positive integer. We associate to each propositional variable p and $i \in \{1, \ldots, n\}$ a new propositional variable p_i. We define the propositional formula $(s(A,n))^i$ by induction on the structure of $s(A,n)$ as follows:*

$(p)^i = p_i$ $\quad$ $(\neg p)^i = \neg p_i$
$(B \wedge C)^i = (B)^i \wedge (C)^i$ $\quad$ $(B \vee C)^i = (B)^i \vee (C)^i$

$$(\Diamond_\lambda B)^i = \begin{cases} \bot & \textit{if} \quad \lambda > n \\ (\bigwedge_{j=1}^{n} x_j \leftrightarrow (B)^j) \wedge \sum_{j=1}^{n} x_j \geqslant \lambda & \textit{otherwise} \end{cases}$$

where $x_1, \ldots, x_n$ are n fresh propositional variables.

Similarly to $(A)^i_n$, $(s(A,n))^i$ is not always polynomial in $|A|$. In order to obtain an equivalent polynomial size formula, we only have to proceed, like in the case of S5 (see Definition 4), by introducing the formula $\mathcal{D}(A)$ obtained by substituting inductively the deep modal subformulas by fresh variables:

- $\mathcal{D}(A) = A$ if A does not contain any deep modal subformula;
- $\mathcal{D}(A) = A[\Diamond_{\lambda_1} B_1/x_{\Diamond_{\lambda_1} B_1}, \ldots, \Diamond_{\lambda_k} B_k/x_{\Diamond_{\lambda_k} B_k}] \bigwedge_{i=1}^{k} \mathcal{D}(\Diamond_1 x_{\Diamond_{\lambda_i} B_i} \to \Diamond_{\lambda_i} B_i)$ $\bigwedge_{i=1}^{k} \mathcal{D}(\Diamond_{\lambda_i} B_i \to \neg\Diamond_1 \neg x_{\Diamond_{\lambda_i} B_i})$.

Proposition 3. *A is GS5 satisfiable in a modal model of n worlds iff $(\mathcal{D}(s(A,n)))^1$ is satisfiable.*

Proof. Similar to the proof of Proposition 1.

As a consequence of Proposition 3 and Theorem 3, we obtain the following theorem:

Theorem 4. *A is GS5 satisfiable iff $\bigvee_{1 \leqslant i \leqslant |A|} (s(A,i))^1$ is satisfiable.*

4 Itemset Support Satisfiability

4.1 Preliminary

Let $\mathcal{I}$ be a finite set of *items*, a set $I \subseteq \mathcal{I}$ is called an *itemset.* A *transaction* is a couple (tid, I) where *tid* is a *transaction identifier* and I is an itemset. A *transaction database* $\mathcal{D}$ is a finite set of transactions over $\mathcal{I}$ where for all two different transactions, they do not have the same identifier. We say that a transaction (tid, I) *supports* an itemset J if $J \subseteq I$.

The *support* of an itemset I in $\mathcal{D}$ is defined by: $\mathcal{S}(I, \mathcal{D}) = |\{tid \mid (tid, J) \in \mathcal{D}\ \texttt{and}\ I \subseteq J\}|$. Moreover, the frequency of I in $\mathcal{D}$ is defined by:
$\mathcal{F}(I, \mathcal{D}) = \frac{\mathcal{S}(I,\mathcal{D})}{|\mathcal{D}|}$.

Let us consider the following example of transaction database over the set of items $\mathcal{I} = \{Spaghetti, Tomato, Parmesan, Beef, Olive\ oil\}$:

tid	itemset
1	*Spaghetti, Tomato, Olive oil*
2	*Spaghetti, Parmesan, Olive oil*
3	*Spaghetti, Olive oil*
4	*Salad, Olive oil*
5	*Spaghetti, Beef, Olive oil*

Transaction database $\mathcal{D}$

For instance, we have $\mathcal{S}(\{Spaghetti, Olive\ oil\}, \mathcal{D}) = |\{1,2,3,5\}| = 4$ and $\mathcal{F}(\{Spaghetti, Olive\ oil\}, \mathcal{D}) = \frac{4}{5}$.

It is easy to see that a transaction databases can be considered as a modal model where the items corresponds to propositional variables and the transactions to worlds. For example, the transaction database in the previous example corresponds to the model $\mathcal{M} = (\{1,2,3,4,5\}, V)$ where $(w, V(w)) \in \mathcal{D}$ (e.g. $V(1) = \{Spaghetti, Tomato, Olive\ oil\}$). We denote by $\mathcal{M}_\mathcal{D}$ (resp. $\mathcal{D}_\mathcal{M}$) the modal model (resp. the transaction database) associated to the transaction database (resp. the modal model) $\mathcal{D}$ (resp. $\mathcal{M}$).

Using the fact that each transaction database has a corresponding modal model, one can encode some transaction database problems in $\mathsf{GS5}$. Thus, the verification techniques in $\mathsf{GS5}$, in particular using SAT, can be used to prove properties on such problems. For instance, let us consider the problem of computing frequent itemsets which is one of the most studied problems in data mining. Let $\mathcal{D}$ be a transaction database over $\mathcal{I}$ and λ a minimal support threshold. The frequent itemset mining problem consists in computing the following set:

$$\mathcal{FIM}(\mathcal{D}, \lambda) = \{I \subseteq \mathcal{I} \mid \mathcal{S}(I, \mathcal{D}) \geqslant \lambda\}$$

It can be encoded in $\mathsf{GS5}$ as follows: $I \in \mathcal{FIM}(\mathcal{D}, \lambda) \Leftrightarrow \mathcal{M}_\mathcal{D} \vDash \Diamond_\lambda \bigwedge_{i \in I} i$. For example, in order to prove that for all transaction database, for all minimal support threshold λ and for all itemsets I and J, if $I \subseteq J$ and $J \in \mathcal{FIM}(\mathcal{D}, \lambda)$ then $I \in \mathcal{FIM}(\mathcal{D}, \lambda)$ (anti-monotonicity), it suffices to prove that $F = \Diamond_\lambda(p \wedge q) \rightarrow \Diamond_\lambda p$ is a theorem of $\mathsf{GS5}$. Indeed, if F is not a theorem then there exists a transaction database (a modal model) where the anti-monotonicity property is not true. Otherwise, it suffices to notice that if $I \subseteq J$ then there exists K such that $\bigwedge_{j \in J} j \equiv (\bigwedge_{i \in I} i) \wedge (\bigwedge_{k \in K} k)$. Since $\mathcal{M}_\mathcal{D} \vDash \Diamond_\lambda \bigwedge_{j \in J} j$ and F is a theorem of $\mathsf{GS5}$, we deduce that $\mathcal{M}_\mathcal{D} \vDash \Diamond_\lambda \bigwedge_{i \in I} i$ and then $I \in \mathcal{FIM}(\mathcal{D}, \lambda)$.

4.2 $\mathsf{SUPPSAT}^n$ Problem

A *support constraint* is an expression of the form $I \lhd k$ where I is a an itemset, $\lhd \in \{\leqslant, \geqslant\}$ and k is a natural number. A transaction database $\mathcal{D}$ satisfies a support constraint $I \lhd k$ if and only if $\mathcal{S}(I, \mathcal{D}) \lhd k$.

Problem 1 ($\mathsf{SUPPSAT}^n$*).* Given a set of support constraints $\mathcal{E}$, determine if there exists a database $\mathcal{D}$ with n transaction satisfying all the constraints in $\mathcal{E}$.

Now, let us encode $\mathsf{SUPPSAT}^n$ in $\mathsf{GS5}$ satsifiability (and consequently in SAT).

Definition 8. *Let* $\mathcal{E} = \{I_1 \lhd_1 k_1, \ldots, I_m \lhd_m k_m\}$ *be a set of support constraints. The* $\mathsf{GS5}$ *formula associated to* $\mathcal{E}$*, denoted* $F_\mathcal{E}$*, is* $\bigwedge_{i=1}^m F(I_i \lhd_i k_i)$ *where* $F(I \geqslant k) = \Diamond_k \bigwedge_{j \in I} j$ *and* $F(I \leqslant k) = \neg\Diamond_{k+1} \bigwedge_{j \in I} j$*.*

Proposition 4. *A set of support constraints* $\mathcal{E}$ *is* $\mathsf{SUPPSAT}^n$ *iff* $F_\mathcal{E} \wedge \neg\Diamond_{n+1}\top$ *is* $\mathsf{GS5}$ *satsifiable.*

Proof.
- *Part* $\Rightarrow$. Let $\mathcal{D}$ be a database with n transaction satisfying $\mathcal{E}$, then it is easy to see that $\mathcal{M}_\mathcal{D}$ satisfies $F_\mathcal{E}$. Moreover, since $\mathcal{M}_\mathcal{D}$ contains n worlds, it satisfies also $\neg\Diamond_{n+1}\top$.

- *Part* $\Leftarrow$. Let $\mathcal{M}$ be a model satisfying $F_{\mathcal{E}} \wedge \neg\Diamond_{n+1}\top$. Then $\mathcal{D}_{\mathcal{M}}$ is a database with m transactions satisfying $\mathcal{E}$ where $m \leqslant n$ because of $\neg\Diamond_{n+1}\top$. Let $\mathcal{D}'$ be the transaction database obtained from $\mathcal{D}$ by adding $n - m$ empty transactions (transaction without items). It is trivial that $\mathcal{D}'$ is a database with n transactions satisfying $\mathcal{E}$.

5 Conclusion and Perspectives

In this work, we study the satisfiability problem in the modal logic S5 and its graded version GS5. We provide reductions of S5 satsifiability and GS5 satisfiability to propositional satisfiability (SAT). The key point is that a modal formula is satisfiable in S5 or GS5 if and only if it is satisfiable in a model with at most the number of the modal subformulas as the number of worlds. Moreover, in order to show how GS5 satisfiablity can be used in data mining, we consider a satisfiability problem related to the known frequent itemset mining problem, called itemset support satisfiability and denoted by $\mathsf{SUPPSAT}^n$. We show how $\mathsf{SUPPSAT}^n$ can be encoded simply in GS5 satisfiability.

In further works we will study the possibility to develop a suitable resolution method for GS5 satisfiability problem. In this context, we think that we have to find a useful form for the modal formulas similar to conjunctive normal form in the propositional case. We also have to propose additional rules to deal with the modal connectives. Such a method can be more appropriate for GS5 satisfiability and also for itemset support satisfiability than a reduction to SAT. Moreover, we will study the satisfiability problem in some extensions of GS5 with new connectives, for example, with the connective $>>$ where $A >> B$ means that the number of worlds satisfying A are greater than the number of those satisfying B. Such a connective will allow us to reason on relations between property supports.

Acknowledgment. We thank the CNRS (Centre National de la Recherche Scientifique) and the ANR (Agence Nationale de la Recherche) for supporting this work.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: SIGMOD, pp. 207–216. ACM Press (1993)
2. Areces, C., Hoffmann, G., Denis, A.: Modal logics with counting. In: Dawar, A., de Queiroz, R. (eds.) WoLLIC 2010. LNCS, vol. 6188, pp. 98–109. Springer, Heidelberg (2010)
3. Baader, F., Buchheit, M., Hollunder, B.: Cardinality restrictions on concepts. Artificial Intelligence 88(1-2), 195–213 (1996)
4. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press (2001)
5. Calders, T.: Axiomatization and Deduction Rules for the Frequency of Itemsets. PhD Thesis. Universiteit Antwerpen (2003)

6. Calders, T.: Itemset frequency satisfiability: Complexity and axiomatization. Theoretical Computer Science 394(1-2), 84–111 (2008)
7. Fine, K.: Cut-free modal sequents for normal modal logics. Notre-Dame Journal of Formal Logic 13(4), 516–520 (1972)
8. Fitting, M.: Proof methods for modal and intuitionistic logics. Synthese Library, vol. 169. Kluwer (1983)
9. Fitting, M.: A simple propositional S5 tableau system. the Annals of Pure and Applied Logic 96, 107–115 (1999)
10. Giunchiglia, E., Giunchiglia, F., Sebastiani, R., Tacchella, A.: Sat vs. Translation Based decision procedures for modal logics: a comparative evaluation. Journal of Applied Non-Classical Logics 10(2) (2000)
11. Giunchiglia, E., Tacchella, A., Giunchiglia, F.: Sat-based decision procedures for classical modal logics. J. Autom. Reasoning 28(2), 143–171 (2002)
12. Warners, J.P.: A linear-time transformation of linear inequalities into conjunctive normal form. Information Processing Letters (1996)
13. Ladner, R.E.: The computational complexity of provability in systems of modal propositional logic. SIAM Journal Computation 6(3), 467–480 (1977)
14. Mielikäinen, T.: On Inverse Frequent Set Mining. In: 2nd IEEE ICDM Workshop on Privacy Preserving Data Mining (PPDM), pp. 18–23. IEEE (2003)
15. Negri, S.: Proof analysis in modal logic. Journal of Philosophical Logic 34, 507–534 (2005)
16. Ohlbach, H.J.: Semantics-based translation methods for modal logics. J. Log. Comput. 1(5), 691–746 (1991)
17. Sebastiani, R., McAllester, D.: New upper bounds for satisfiability in modal logics – the case-study of modal K (1997)
18. Marques-Silva, J., Lynce, I.: Towards robust CNF encodings of cardinality constraints. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 483–497. Springer, Heidelberg (2007)
19. Sinz, C.: Towards an optimal CNF encoding of boolean cardinality constraints. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 827–831. Springer, Heidelberg (2005)
20. Tseitin, G.S.: On the complexity of derivations in the propositional calculus. In: Slesenko, H.A.O. (ed.) Structures in Constructives Mathematics and Mathematical Logic, Part II, pp. 115–125 (1968)
21. van der Hoek, W., de Rijke, M.: Counting Objects. Journal of Logic and Computation 5(3), 325–345 (1995)
22. Wansing, H.: Sequent systems for modal logics. In: Gabbay, D., Guenther, F. (eds.) Handbook of Philosophical Logic, 2nd edn., vol. 8, pp. 61–145. Kluwer, Dordrecht (2002)

Author Index